Jaime Simão Sichman
Helder Coelho
Solange Oliveira Rezende (Eds.)

LNAI 4140

Advances in Artificial Intelligence – IBERAMIA-SBIA 2006

2nd International Joint Conference:
10th Ibero-American Conference on AI
18th Brazilian AI Symposium
Ribeirão Preto, Brazil, October 2006, Proceedings

Springer

Lecture Notes in Artificial Intelligence 4140

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Jaime Simão Sichman Helder Coelho
Solange Oliveira Rezende (Eds.)

Advances in Artificial Intelligence – IBERAMIA-SBIA 2006

2nd International Joint Conference:
10th Ibero-American Conference on AI
18th Brazilian AI Symposium
Ribeirão Preto, Brazil, October 23-27, 2006
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Jaime Simão Sichman
University of São Paulo, Computer Engineering Department
Av. Prof. Luciano Gualberto, tv. 3, 158, 05508-970 São Paulo, SP, Brazil
E-mail: jaime.sichman@poli.usp.br

Helder Coelho
University of Lisbon, Department of Informatics
Edifício C6, Campo Grande, 1749-016 Lisboa, Portugal
E-mail: hcoelho@di.fc.ul.pt

Solange Oliveira Rezende
University of São Paulo, Department of Computer Science
Av. Trabalhador São-Carlense, 400, 13560-970 São Carlos, SP, Brazil
E-mail: solange@icmc.usp.br

Library of Congress Control Number: 2006933826

CR Subject Classification (1998): I.2.6-7, I.2, F.1.1, F.4.2-3, F.2

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-45462-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-45462-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11874850 06/3142 5 4 3 2 1 0

Preface

The Brazilian Artificial Intelligence (AI) community decided in 2004 to organize an International Joint Conference, joining IBERAMIA 2006 (the 10th Ibero-American Artificial Intelligence Conference), SBIA 2006 (the 18th Brazilian Artificial Intelligence Symposium), and SBRN 2006 (the 9th Brazilian Neural Networks Symposium). This decision was a consequence of the successful event organized in 2000, when the First International Joint Conference IBERAMIA/ SBIA 2000 (7th Ibero-American Artificial Intelligence Conference and 15th Brazilian Artificial Intelligence Symposium) occurred in Brazil. Moreover, in 2006 the artificial intelligence community celebrated the golden anniversary of the 1956 Dartmouth Conference that marked the beginning of artificial intelligence as a research field.

SBIA 2006 was the 18^{th} conference of the SBIA conference series, which is the leading Brazilian conference for the presentation of AI research and applications. Since 1995, SBIA has become an international conference, with papers written in English, an international Program Committee, and proceedings published in Springer's *Lecture Notes in Artificial Intelligence* (LNAI) series.

IBERAMIA 2006 was the 10^{th} conference of the IBERAMIA conference series, which has been one of the most suitable forums for Ibero-American AI researchers (from South and Central American countries, Mexico, Spain and Portugal) to present their results. Following the SBIA and EPIA (Portuguese Conference on AI) experiences, from IBERAMIA 1998 on, it has also become an international conference with proceedings published in Springer's LNAI series.

The International Joint Conference was held in Ribeirão Preto, Brazil, during October 23-27, 2006. The call for papers was very successful. In the registration phase, when the authors were asked to submit just the title and abstract of their papers, 281 submissions were received. In the second phase, 246 authors uploaded their full paper to be reviewed. These submissions came from 23 different countries[1], as shown in Table 1. After the revision process, 62 papers were accepted to be published in these proceedings.

In order to improve the overall quality of this joint conference, a double-blind reviewing procedure was adopted, and the acceptance rate was fixed around 25% of the submitted papers. A large group of reviewers from all content areas was set up to carry out the difficult and challenging work of selecting the papers. Every paper was evaluated by at least three reviewers, and ranked by the JEMS system. Whenever conflicts were found, open discussions among the reviewers were triggered and analyzed later to support our final decision.

The conference included keynote speeches, introductory and advanced tutorials by world-leading researchers, several workshops covering specific topics and a thesis and dissertation contest.

[1] In the case of several authors from different countries, we have considered the first author affiliation.

Table 1. IBERAMIA/SBIA 2006 Paper Submission Summary

Country	Registered	Submitted	Accepted
Argentina	15	14	3
Australia	1	1	1
Brazil	149	128	33
Chile	8	8	2
China	3	3	–
Colombia	1	–	–
Cuba	3	3	3
Ecuador	2	2	–
France	9	9	2
Germany	2	2	–
Italy	1	1	1
Korea	5	5	–
Malaysia	1	1	–
Mexico	44	37	8
Poland	2	2	–
Portugal	8	5	1
Spain	18	17	6
Taiwan	1	1	–
Turkey	2	2	–
UK	1	1	–
Uruguay	1	1	1
USA	1	–	–
Venezuela	3	3	1
Total	281	246	62

We would like to thank all the authors for submitting their papers, as well as all members of the Program Committee and the additional referees for their hard work. The high quality of the papers included in this volume would not be possible without their participation and diligence. We also gratefully acknowledge the help of our colleagues whose support in the organization of this conference was invaluable. Finally, we would like to express our gratitude to Alfred Hofmann and his staff at Springer for giving us again the opportunity to publish the proceedings in the LNAI series.

October 2006

Jaime Simão Sichman
Helder Coelho
Solange Oliveira Rezende

Organization

The International Joint Conference IBERAMIA/SBIA/SBRN 2006 was organized by several AI research groups that belong to the University of São Paulo (USP), Brazil:

1. Instituto de Ciências Matemáticas e de Computação (ICMC)
   Departamento de Ciências de Computação (SCE)
   Laboratório de Inteligência Computacional (LABIC)
   Campus São Carlos
2. Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP)
   Departamento de Física e Matemática (DFM)
   Campus Ribeirão Preto
3. Laboratório de Técnicas Inteligentes (LTI)
   Departamento de Engenharia de Computação e Sistemas Digitais (PCS)
   Escola Politécnica (EP)
   Campus São Paulo
4. Instituto de Matemática e Estatística (IME)
   Departamento de Ciência da Computação (MAC)
   Laboratório de Lógica, Inteligência Artificial e Métodos Formais (LIAMF)
   Campus São Paulo
5. Escola de Artes, Ciências e Humanidades (EACH)
   Curso de Sistemas de Informação (SI)
   Campus São Paulo

Organizing Committee

General Chair

Solange Oliveira Rezende (LABIC/SCE/ICMC)

General Co-chair

Antonio Carlos Roque da Silva Filho (DFM/FFCLRP)

Program Chairs

Jaime Simão Sichman (LTI/PCS/EP, Brazil)
Helder Coelho (LABMAG/DI/FC/UL, Portugal)

Workshop Chair

Maria das Graças Volpe Nunes (LABIC/SCE/ICMC)

Tutorial Chair

Maria Carolina Monard (LABIC/SCE/ICMC)

Local Organization Committee

Alneu de Andrade Lopes (LABIC/SCE/ICMC)
Flavio Soares Correa da Silva (LIAMF/MAC/IME)
Ivandré Paraboni (SI/EACH)
José de Jesus Perez Alcazar (SI/EACH)
Leliane Nunes de Barros (LIAMF/MAC/IME)
Marcelo Finger (LIAMF/MAC/IME)
Renato Tinos (DFM/FFCLRP)
Roseli Aparecida Francelim Romero (LABIC/SCE/ICMC)
Sandra Maria Aluisio (LABIC/SCE/ICMC)
Thiago Alexandre Salgueiro Pardo (LABIC/SCE/ICMC)
Zhao Linag (LABIC/SCE/ICMC)

IBERAMIA Steering Committee

Alvaro Albornoz (ITESM, Mexico)
Arlindo Oliveira (INESC, Portugal)
Carlos Alberto Reyes (INAOE, Mexico)
Christian Lemaitre (UAM, Mexico)
Federico Barber (UPV, Spain)
Francisco Garijo (TI+D, Spain)
Helder Coelho (UL, Portugal)
Jaime Simão Sichman (USP, Brazil)
Miguel Toro (US, Spain)

SBIA Steering Committee

Ana Bazzan (UFRGS, Brazil)
Geber Ramalho (UFPE, Brazil)
Gerson Zaverucha (UFRJ, Brazil)
Guilherme Bittencourt (UFSC, Brazil)
Maria Carolina Monard (USP, Brazil)

Supporting Scientific Societies

SBC	Sociedade Brasileira de Computação
AEPIA	Asociación Española para Inteligencia Artificial
APPIA	Associação Portuguesa para Inteligência Artificial
SMIA	Sociedad Mexicana de Inteligencia Artificial

Sponsoring Institutions

The International Joint Conference 2006 had the financial support of the following Brazilian institutions:

FAPESP	Fundação de Amparo à Pesquisa do Estado de São Paulo
CNPq	Conselho Nacional de Pesquisa
CAPES	Coordenação de Aperfeiçoamento do Pessoal de Nível Superior
FINEP	Financiadora de Estudos e Projetos

Program Committee

Adolfo Guzmán-Arenas	Instituto Politécnico Nacional (Mexico)
Adina Florea	University Politehnica of Bucharest (Romania)
Alberto Oliart	Instituto Tecnológico y de Estudios Superiores de Monterrey (Mexico)
Aldo Dragoni	Università Politecnica delle Marche (Italy)
Alejandro Zunino	Universidad Nac. del Centro de la Pcia. de Buenos Aires (Argentina)
Alexander Gelbukh	Instituto Politécnico Nacional (Mexico)
Alexis Drogoul	Institut de Recherche pour le Développement (France)
Alipio Jorge	Universidade do Porto (Portugal)
Alvaro Albornoz	Instituto Tecnológico y de Estudios Superiores de Monterrey (Mexico)
Amal El Fallah Seghrouchni	Université Paris VI (France)
Ana Bazzan	Universidade Federal do Rio Grande do Sul (Brazil)
Ana Paiva	Universidade Técnica de Lisboa (Portugal)
Ana Serrano	Universidad Nacional EAD (Spain)
Ana Cristina Garcia	Universidade Federal Fluminense (Brazil)
Ana Teresa Martins	Universidade Federal do Ceará (Brazil)
Analia Amandi	Universidad Nac. del Centro de la Pcia. de Buenos Aires (Argentina)

André Carvalho	Universidade de São Paulo (Brazil)
Andre Valente	Knowledge Systems Ventures LLC (USA)
Andrea Omicini	Università di Bologna (Italy)
Angel Kuri	Instituto Tecnológico Autónomo de México (Mexico)
Anna Helena Reali Costa	Universidade de São Paulo (Brazil)
Antônio Rocha Costa	Universidade Católica de Pelotas (Brazil)
Ariadne Carvalho	Universidade de Campinas (Brazil)
Arlindo Oliveira	Instituto de Engenharia de Sistemas e Computadores (Portugal)
Arturo Hernandez Aguirre	Centro de Investigación en Matemáticas (Mexico)
Augustin Lux	Institut National de Recherche en Informatique et en Automatique (France)
Augusto Loureiro da Costa	Universidade Federal da Bahia (Brazil)
Brian Mayoh	Aarhus University (Denmark)
Carles Sierra	Institut d'Investigación en Intelligència Artificial (Spain)
Carlos Alberto Reyes	Instituto Nacional de Astrofísica Óptica y Electrónica (Mexico)
Carlos Brizuela	Centro de Investigación Científica y de Educación Superior de Enseada (Mexico)
Carlos Castro	Universidad Técnica Federico Santa María (Chile)
Carlos Coello	Instituto Politécnico Nacional (Mexico)
Carlos Ribeiro	Instituto Tecnológico da Aeronáutica (Brazil)
Catherine Tessier	Office National d'Etudes et de Recherches Aérospatiales (France)
Catholijn Jonker	Nijmegen Institute for Cognition and Information (The Netherlands)
Celso Kaestner	Pontifícia Universidade Católica-PR (Brazil)
Christian Lemaitre	Universidad Autónoma Metropolitana (Mexico)
Cora Toledo	Laboratorio Nacional de Informática Avanzada (Mexico)
Cristiano Castelfranchi	Istituto di Scienze e Tecnologie della Cognizione (Italy)
David Hales	Universitá di Bologna (Italy)

Eduardo Fermé	Universidade da Madeira (Portugal)
Eduardo Gomez-Ramirez	Universidad La Salle (Mexico)
Eduardo Morales	Instituto Nacional de Astrofísica Óptica y Electrónica (Mexico)
Elizabeth Sklar	City University of New York (USA)
Emilio Hernandez	Universidade de São Paulo (Brazil)
Emma Norling	Centre for Policy Modelling (UK)
Eric Matson	Wright State University (USA)
Ernesto Costa	Universidade de Coimbra (Portugal)
Estela Bicho	Universidade do Minho (Portugal)
Eugenio Oliveira	Universidade do Porto (Portugal)
Evandro Costa	Universidade Federal de Alagoas (Brazil)
Fábio Cozman	Universidade de São Paulo (Brazil)
Federico Barber	Universidad Politécnica de Valencia (Spain)
Fernando Carvalho	Universidade Federal do Ceará (Brazil)
Fernando Osório	Universidade do Vale do Rio dos Sinos (Brazil)
Flavia Barros	Universidade Federal de Pernambuco (Brazil)
Flávio Soares Corrêa da Silva	Universidade de São Paulo (Brazil)
Flávio Varejão	Universidade Federal do Espírito Santo (Brazil)
François Jacquenet	Université Jean Monnet (France)
Francisco Garijo	Telefonica I+D (Spain)
Fréderic Amblard	Université Toulouse I (France)
Fred Freitas	Universidade Federal de Pernambuco (Brazil)
Gabriel Pereira Lopes	Universidade Nova de Lisboa (Portugal)
Gabriela Henning	Universidad Nacional del Litoral (Argentina)
Gabriela Lindemann	Humboldt University (Germany)
Geber Ramalho	Universidade Federal de Pernambuco (Brazil)
Gerson Zaverucha	Universidade Federal do Rio de Janeiro (Brazil)
Gina Oliveira	Universidade Federal de Uberlândia (Brazil)
Graça Gaspar	Universidade de Lisboa (Portugal)
Graça Marietto	Universidade do Estado de Santa Catarina (Brazil)
Guido Boella	Universitá di Torino (Italy)
Guilherme Bittencourt	Universidade Federal de Santa Catarina (Brazil)
Guillermo Simari	Universidad Nacional del Sur (Argentina)
Gustavo Giménez-Lugo	Centro Universitário Positivo (Brazil)

Harko Verhagen	Stockholm University (Sweden)
Hector Geffner	Universitat Pompeu Fabra (Spain)
Helder Coelho	Universidade de Lisboa (Portugal) **(Chair)**
Heloisa Camargo	Universidade Federal de São Carlos (Brazil)
Hugo Terashima-Marín	Instituto Tecnológico y de Estudios Superiores de Monterrey (Mexico)
Ioan Letia	Tech. University of Cluj-Napoca (Romania)
Jacinto Davila	Universidad de los Andes (Colombia)
Jacques Wainer	Universidade de Campinas (Brazil)
Jaime Simão Sichman	Universidade de São Paulo (Brazil) **(Chair)**
Jeffrey Rosenchein	Hebrew University of Jerusalem (Israel)
João Gama	Universidade do Porto (Portugal)
John Atkinson	Universidad de Concepcion (Chile)
Jomi Fred Hubner	Universidade Regional de Blumenau (Brazil)
Jordi Sabater	Institut d'Investigación en Intelligència Artificial (Spain)
José Reinaldo Silva	Universidade de São Paulo (Brazil)
Juan Pavón	Universidad Complutense Madrid (Spain)
Juan Sossa Azuela	Instituto Politécnico Nacional (Mexico)
Juan M. Corchado	Universidad de Salamanca (Spain)
Keiki Takadama	Tokyo Institute of Technology (Japan)
Krzystof Cetnarowicz	University of Krakow (Poland)
Lúcia Giraffa	Pontifícia Universidade Católica-RS (Brazil)
Leliane Nunes de Barros	Universidade de São Paulo (Brazil)
Leonardo Garrido	Instituto Tecnológico y de Estudios Superiores de Monterrey (Mexico)
Liz Sonenberg	University of Melbourne (Australia)
Luís Correia	Universidade de Lisboa (Portugal)
Luís Paulo Reis	Universidade do Porto (Portugal)
Luciano Garrido	Universidad de Havana (Cuba)
Luis Morgado	Instituto Politécnico de Lisboa (Portugal)
Luis Pineda	Universidad Nacional Autónoma de México (Mexico)
Luis Sucar	Instituto Nacional de Astrofísica Óptica y Electrónica (Mexico)
Luis Otávio Alvares	Universidade Federal do Rio Grande do Sul (Brazil)
Marcelo Finger	Universidade de São Paulo (Brazil)
Marcelo Ladeira	Universidade de Brasilia (Brazil)

Ramon Brena	Instituto Tecnológico y de Estudios Superiores de Monterrey (Mexico)
Ramon Mantaras	Institut d'Investigación en Intelligència Artificial (Spain)
Ramon Otero	Universidad da Coruña (Spain)
Reinaldo Bianchi	Centro Universitario da FEI (Brazil)
Renata Vieira	Universidade do Vale do Rio dos Sinos (Brazil)
Renata Wassermann	Universidade de São Paulo (Brazil)
Ricardo Contreras A.	Universidad de Concepción (Chile)
Rino Falcone	Istituto di Scienze e Tecnologie della Cognizione (Italy)
Roque Morales	Universidad de Murcia (Spain)
Rosa Viccari	Universidade Federal do Rio Grande do Sul (Brazil)
Roseli Francelin Romero	Universidade de São Paulo (Brazil)
Salvador Abreu	Universidade de Evora (Portugal)
Sandra Sandri	Institut d'Investigación en Intelligència Artificial (Spain)
Sascha Ossowski	Universidad Rey Juan Carlos (Spain)
Sefer Kurnaz	Turkish Air Force Academy (Turkey)
Sheila Veloso	Universidade Federal do Rio de Janeiro (Brazil)
Syeda Fatima	Osmania University (India)
Teresa Ludermir	Universidade Federal de Pernambuco (Brazil)
Uma Garimella	AP State Council for Higher Education (India)
Vera Strube de Lima	Pontifícia Universidade Católica-RS (Brazil)
Veronica Dahl	Simon Fraser University (Canada)
Vicente Botti	Universidad Politécnica de Valencia (Spain)
Virginia Dignum	University of Utrecht (The Netherlands)
Wamberto Vasconcelos	University of Aberdeen (UK)

Additional Reviewers

Adrian Perreau de Pinninck	Alvaro Moreira
Adriana Giret	Ana Lorena
Alberto Fernández	Ana Luisa Respicio
Alessandro Safiotti	Andreia Machion
Alexandre Cavalcante	António Abreu
Alexandre Simões	Armando Sousa

Marcio Netto	Universidade de São Paulo (Brazil)
Marco Dorigo	Université Libre de Bruxelles (Belgium)
Maria Cravo	Universidade Técnica de Lisboa (Portugal)
Maria Carolina Monard	Universidade de São Paulo (Brazil)
Maria das Graças Volpe Nunes	Universidade de São Paulo (Brazil)
Maria do Carmo Nicoletti	Universidade Federal de São Carlos (Brazil)
Maria-Cristina Riff	Universidad Técnica Federico Santa María (Chile)
Marie-Pierre Gleizes	Institut de Recherche en Informatique de Toulouse (France)
Mario Campos	Universidade Federal de Minas Gerais (Brazil)
Matthew Turk	University of California at Santa Barbara (USA)
Mauricio Galindo	Fundacion Universidad de las Americas (Mexico)
Mehdi Dastani	University of Utrecht (The Netherlands)
Michail Kalogiannakis	Technological Educational Institute of Crete (Greece)
Miguel Salido	Universidad Politécnica de Valencia (Spain)
Miguel Toro	Universidad de Sevilla (Spain)
Nuno David	Instituto Superior de Ciências do Trabalho e da Empresa (Portugal)
Olac Fuentes	Instituto Nacional de Astrofísica Óptica y Electrónica (Mexico)
Olivier Boissier	Ecole Nationale Supérieure des Mines de Saint-Etienne (France)
Onn Shehory	IBM - Haifa Research Labs (Israel)
Osvaldo Cairo	Instituto Tecnológico Autónomo de México (Mexico)
Oswaldo Teran	Universidad Los Andes (Venezuela)
Pablo Noriega	Institut d'Investigación en Intelligència Artificial (Spain)
Paul Davidsson	Blekinge Institute of Technology (Sweden)
Paulo Quaresma	Universidade de Évora (Portugal)
Pavel Brazdil	Universidade do Porto (Portugal)
Pedro Barahona	Universidade de Nova Lisboa (Portugal)
Pedro de Oliveira	Universidade Presbiteriana Mackenzie (Brazil)
Rafael Bordini	University of Durham (UK)
Ramiro Neves	Universidade Técnica de Lisboa (Portugal)

Arnaldo Albuquerque
Bengt Carlsson
Birna van Riemsdijk
Broderick Crawford
Bruno de Souza
Carlos Chesevar
Carlos Grilo
Carlos Lourenço
Carlos Augusto Prolo
Cristian Mateos Diaz
Daniela Godoy
Dibio Leandro Borges
Didac Busquets
Dietmar Wolz
Eduardo Spinosa
Eric Monfroy
Estevam Rafael Hruschka Jr.
Esther Colombini
Fabio Okuyama
Francisco Reinaldo
Francisco Silva
George Cavalcanti
Gustavo Batista
Hae Yong Kim
Helio Barbosa
Hugo Marques
Jan Broersen
Jeronimo Pellegrini
João Alcântara
João Balsa
Jorge Jambeiro
Jorge J. Gómez-Sanz
José Cruz
José Carlos Mombach
José Luiz Pio
Jose Torres
Ligia Ferreira
Luís Moniz
Luciano Pimenta
Luis Rueda
Luiz Lorena
Marcelo Armentano
Marcelo Blois
Marcia Aparecida
Marco Schorlemmer
Matteo Baldoni
Mêuser Valença
Miguel Vázquez
Norton Trevisan Roman
Olivier Perrin
Orides Morandin Junior
Oscar Sapena
Paolo Remagnino
Patrícia Tedesco
Patricia Shaw
Paula Brito
Paulo Urbano
Paulo Roberto Ferreira Jr.
Pedro Faria
Pedro Mariano
Pedro Santana
Phil Husbands
Rafael Parpinelli
Ricardo Rodriguez
Ricardo Tanscheit
Ricardo Zebulum
Robinson Noronha
Ronaldo Prati
Ronaldo Teixeira Martins
Rubén Fuentes
Sergio Gomez
Shervin Nouyan
Silvia Moraes
Silvia Schiaffino
Silvio Pereira
Siome Goldenstein
Stefan Johansson
Teresa Gonçalves
Thiago Pardo
Tsang Ing Ren
Valdinei Silva
Valguima Odakura
Vera Lúcia da Silva
Vitor Nogueira

Table of Contents

Invited Speakers

AI in Education and Intelligent Tutoring Systems

Autonomous Agents and Multiagent Systems

Computer Vision and Pattern Recognition

Evolutionary Computation and Artificial Life

Hybrid Systems (Fuzzy, Genetic, Neural, Symbolic)

Knowledge Acquisition and Machine Learning

Knowledge Discovery and Data Mining

Knowledge Engineering, Ontologies and Case Based Reasoning

Knowledge Representation and Reasoning

Neutral Language Processing

Planning and Scheduling

Robotics

Theoretical and Logical Methods

Uncertainty

Organizing Software Agents

Frank Dignum

Department of Information and Computing Sciences
Universiteit Utrecht, The Netherlands

In recent years the field of multi-agent systems (MAS) research has become mature. It has become apparent that the use of the MAS paradigm facilitates the modularization of complex systems in autonomous subsystems (implemented as agents) and their interactions. However, most systems that are implemented have either a very strict central control or have no central organization at all. The first type of systems use the MAS paradigm mainly to distribute the computation power. The outcomes of the system are in principle analytically determinable, but little use is made of the autonomy of the agents. The second type of systems take agents as autonomous entities pursuing their own individual goals based on their own beliefs and capabilities. In this perspective, global behavior emerges from individual interactions and therefore the final behavior of the whole system cannot be predicted, easily managed or specified externally.

We are trying to provide a middle ground by modelling MAS as agent *organizations*. The definition of the organization of which the agents form a part provides the framework and constraints within which the agents will operate. The organizational specification should therefore ensure some desirable global behavior of the system. At the same time the specification of the organizational framework leaves the agents fulfilling positions in this organization enough autonomy to react to unforeseen circumstances and interactions and therefore optimize operational behavior of the system.

The organizational framework is specified on different levels of abstraction and along different dimensions. It is illustrated in the following figure.

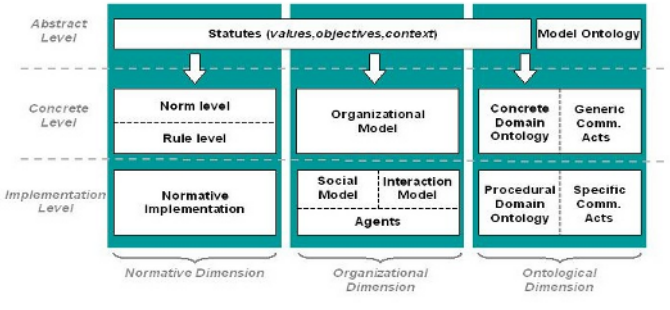


Fig. 1. Levels and dimensions in the organizational framework

The levels of abstraction are used to separate different types of concerns. At the most abstract level we specify what are the global goals (objectives) and

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 1–2, 2006.

required outcomes of the system. We can also specify which constraints are put on the system by the context in which it operates.

At the concrete level we specify the conceptual models of the organization. Along the normative dimension, we specify which are the norms that govern the behavior of the agents within the organization and give indications about norm enforcement mechanisms. It is important to talk about *norms* instead of the more traditional *constraints*, because we assume that (some) norms can be violated by the agents. E.g. if an agent has to release a resource after a certain amount of time we cannot force the agent to actually do this. We do, however, specify the punishment when a norm is violated.

The central concept of the organizational model is the *role*. The role is used in terms of a position that can be filled by agents in the implementation. The organizational model specifies how the global goals are divided over different roles within an organization and how the agents that will fill the roles can interact in order to achieve the goals. The interactions between the roles are specified at a high level. We are mainly concerned about ensuring a certain outcome of the interaction, not determining exactly which protocol should be used at this level.

The ontological dimension on this level specifies the terms used in communication and which message types are used to realize the communication in the interaction between agents.

At the implementation level we determine which mechanisms will be used to enforce norms, which protocols should be used for certain interactions (possibly using standard FIPA protocols), etc. This is also the level where we determine how agents can fulfill roles. I.e. do the agents have their own goals or just the goals from the roles they fulfill, can they change roles and/or enact different roles at the same time, etc.

The main advantages of using this organizational framework when designing MAS are:

1. Using roles as intermediary concepts between global goals and individual agents gives the opportunity to balance between central control and local autonomy. Besides full central control and full local autonomy, in our approach we can also specify mixtures of these two extremes.
2. Using norms in the early phases of the design leaves open possibilities to implement them while making designers aware that they also have to provide means to enforce them.

More details of the framework can be found in [2,1]. Here we provide examples and a theoretical foundation for this work and a first sketch of a methodology.

References

1. V. Dignum. *A Model for Organizational Interaction: based on Agents, founded in Logic*. SIKS Dissertation Series 2004-1. SIKS, 2004. PhD Thesis.
2. J. Vazquez-Salceda, V. Dignum, and F.Dignum. Organizing multiagent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 11(3):307–360, 2005.

Learning, Logic, and Probability: A Unified View

Pedro Domingos

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195, U.S.A.
pedrod@cs.washington.edu
http://www.cs.washington.edu/homes/pedrod

AI systems must be able to learn, reason logically, and handle uncertainty. While much research has focused on each of these goals individually, only recently have we begun to attempt to achieve all three at once. In this talk, I describe Markov logic, a representation that combines first-order logic and probabilistic graphical models, and algorithms for learning and inference in it. Syntactically, Markov logic is first-order logic augmented with a weight for each formula. Semantically, a set of Markov logic formulas represents a probability distribution over possible worlds, in the form of a Markov network with one feature per grounding of a formula in the set, with the corresponding weight. Formulas are learned from relational databases using inductive logic programming techniques. Weights can be learned either generatively (using pseudo-likelihood optimization) or discriminatively (using a voted perceptron algorithm). Inference is performed by a weighted satisfiability solver or by Markov chain Monte Carlo, operating on the minimal subset of the ground network required for answering the query. Experiments in link prediction, entity resolution and other problems illustrate the promise of this approach.

This work, joint with Stanley Kok, Hoifung Poon, Matthew Richardson, and Parag Singla, is described in further detail in Domingos et al. [1]. An open-source implementation of Markov logic and the algorithms described in this talk is available in the Alchemy package [2].

References

1. Domingos, P., Kok, S., Poon, H., Richardson, M., & Singla, P.: Unifying logical and statistical AI. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, AAAI Press, Boston, MA, U.S.A. (2006). http://www.cs.washington.edu/homes/pedrod/papers/aaai06c.pdf.
2. Kok, S., Singla, P., Richardson, M., and Domingos, P.: The Alchemy system for statistical relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, U.S.A. (2005). http://www.cs.washington.edu/ai/alchemy.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, p. 3, 2006.

Reinventing Machine Learning with ROC Analysis

Peter A. Flach

Department of Computer Science, University of Bristol, United Kingdom
Peter.Flach@bristol.ac.uk

Receiver Operating Characteristics (ROC) Analysis originated from signal detection theory, as a model of how well a receiver is able to detect a signal in the presence of noise [1,9]. Its key feature is the distinction between hit rate (or true positive rate) and false alarm rate (or false positive rate) as two separate performance measures. ROC analysis has also widely been used in medical data analysis to study the effect of varying the threshold on the numerical outcome of a diagnostic test. It has been introduced to machine learning relatively recently, in response to classification tasks with skewed class distributions or misclassification costs [11,12,5].

ROC analysis is set to cause a paradigm shift in classification-oriented machine learning. Separating performance on classes is almost always a good idea from an analytical perspective. For instance, it can help us to

- understand the behaviour and skew-sensitivity of many machine learning metrics, including rule learning heuristics and decision tree splitting criteria, by plotting their isometrics in ROC space [4,7];
- develop methods and algorithms to improve the Area Under the ROC Curve (AUC) of a model [10,6];
- understand fundamental classifier training algorithms by visualising how they split up ROC space or its unnormalised cousin, PN space.

In this talk I will mainly illustrate the latter point. One elegant example is the separate-and-conquer or sequential covering rule learning algorithm, that can be visualised by a piecewise linear trajectory from the origin of PN space to the top right-hand corner, each segment corresponding to a learned rule [8]. Since there is no guarantee that the resulting piecewise linear curve is convex, it needs to be followed by a post-processing step in which the rules are sorted on decreasing odds ratio. Decision trees, on the other hand, are grown recursively in a divide-and-conquer fashion, which means that the first point of the curve is found somewhere in the middle. Again, a sorting stage is needed to obtain the set of ROC-optimal labellings [2].

Perhaps more surprisingly, the ROC curve of a naive Bayes classifier can also be grown in a divide-and-conquer fashion. With the basic naive Bayes algorithm no sorting is allowed as this requires access to the joint probabilities, which naive Bayes estimates as products of marginals. As a result, naive Bayes ROC curves can exhibit concavities even over the training set. However, by pre-sorting the attribute values we can 'grow' the model in a way similar to decision trees. I will show how this gives rise to several ways to make the classifier less naive. A similar analysis applies to a naive version of nearest-neighbour classification.

Finally, I will discuss some open problems. More work is needed on approximations for more than two classes, where a full ROC analysis is unfeasible (since for c classes it

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 4–5, 2006.

requires $c(c-1)$ dimensions, i.e., the number of different misclassification types). The insistence of ranks rather than scores is another major shortcoming, since classifiers performing equally in terms of ranking may not perform equally in terms of scores [3,13].

References

1. J.P. Egan. *Signal detection theory and ROC analysis*. Academic Press, New York, 1975.
2. C. Ferri, P.A. Flach, and J. Hernández-Orallo. Learning decision trees using the area under the ROC curve. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, pages 139–146, 2002.
3. C. Ferri, P.A. Flach, J. Hernández-Orallo, and A. Senad. Modifying ROC curves to incorporate predicted probabilities. In *Proceedings of the second workshop on ROC analysis in machine learning*. ICML'05 workshop proceedings, 2005.
4. P.A. Flach. The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, pages 194–201, 2003.
5. P.A. Flach. The many faces of ROC analysis in machine learning, 2004. ICML'04 tutorial notes, `www.cs.bris.ac.uk/~flach/ICML04tutorial/`.
6. P.A. Flach and S. Wu. Repairing concavities in ROC curves. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 702–707, 2005.
7. J. Fürnkranz and P.A. Flach. An analysis of rule evaluation metrics. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, pages 202–209, 2003.
8. J. Fürnkranz and P.A. Flach. ROC 'n' rule learning-towards a better understanding of covering algorithms. *Machine Learning*, 58(1):39–77, 2005.
9. D.M. Green and J.A. Swets. *Signal detection theory and psychophysics*. Wiley, New York, 1966.
10. R.C. Prati and P.A. Flach. ROCCER: An algorithm for rule learning based on ROC analysis. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 823–828, 2005.
11. F.J. Provost and T. Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 43–48, 1997.
12. F.J. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.
13. S. Wu and P.A. Flach. A scored AUC metric for model evaluation and selection. In *Proceedings of the second workshop on ROC analysis in machine learning*. ICML'05 workshop proceedings, 2005.

Cocktail Party Processing

DeLiang Wang

Department of Computer Science & Engineering and Center of Cognitive Science
The Ohio State University - Columbus, OH 43210, USA
dwang@cse.ohio-state.edu

The acoustic environment is typically composed of multiple simultaneous events. A remarkable achievement of the auditory system is its ability to disentangle the acoustic mixture and group the sound energy that originates from the same event or source. This process of auditory organization is referred to as auditory scene analysis. The cocktail party problem, or segregation of speech from interfering sounds, has proven to be extremely challenging from the computational standpoint.

In this presentation I describe an auditory scene analysis approach to the cocktail party problem. Our model starts with simulated auditory periphery or cochleagram. A subsequent stage computes mid-level auditory representations, including correlogram and cross-channel correlation. The core of the model performs segmentation and grouping in a two-dimensional time-frequency representation that encodes proximity in frequency and time, periodicity, amplitude modulation (AM), and onset/offset.

For voiced speech segregation, motivated by psychoacoustic observations our system employs different mechanisms for handling resolved and unresolved harmonics. For unresolved harmonics, the model generates segments based on common AM in addition to temporal continuity, and groups segments according to AM rates. For unvoiced speech segregation, our model decomposes the input mixture into contiguous time-frequency segments by analyzing sound onsets and offsets. Grouping of unvoiced segments is based on Bayesian classification of acoustic-phonetic features.

Our model achieves the state-of-the-art performance in speech segregation, and demonstrates the promise of the auditory scene analysis approach to solving the cocktail party problem.

The following papers give detailed information for my presentation (see `http://www.cse.ohio-state.edu/~dwang/`):

- Hu G. and Wang D.L. (2004): "Monaural speech segregation based on pitch tracking and amplitude modulation," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1135–1150.
- Hu G. and Wang D.L. (2006): "Auditory segmentation based on onset and offset analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, in press.
- Wang D.L. and Hu G. (2006): "Unvoiced speech segregation," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. V.953–956.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, p. 6, 2006.

Diagnostic of Programs for Programming Learning Tools

Karina Valdivia Delgado and Leliane Nunes de Barros*

University of São Paulo, Department of Computer Science,
Rua do Matão, 1010, Bloco C. Cidade Universitária. 05508-090.
São Paulo, SP. Brazil

Abstract. It is not easy for a student to develop programming skills and learn how to construct their own problem solving algorithms. Well designed materials and tools can guide programming students knowledge and skill construction. Such tools may allow students to acquire better and faster, the necessary programming skills. In this paper we show the results of some experiments realized on a set of faulty student's programs using PROPAT_DEBUG, an automatic program debugger, based on the Model Based Diagnosis technique of Artificial Intelligence. The results show that during the interactive debugging process it is possible for a student to learn by answering the questions posed by the AI diagnosis system to discriminate its fault hypotheses.

1 Introduction

Model based diagnosis (MBD), also called *diagnosis from the first principles*, can be described as the interaction between observations and predictions [1]. MBD has been typically applied to troubleshooting physical device. On one hand, we have the actual device whose behavior can be observed, on the other hand we have a model of that device (system description) which is used to make predictions about its behavior. Such model typically describes the components of the system, their connections and the behavior of the components. The difference between an observation and a prediction is called a discrepancy. A match between an observation and a prediction is called a corroboration. Both discrepancies and corroborations are used to identify which parts of the device are possibly incorrect [1].

While engineers troubleshoot mechanical or electrical systems to find broken parts by trying to understand the differences between the physical systems and their models, a computer programming teacher tries to understand the differences between the student program code and his intentions (or the problem goals). If we see the student program as the system the teacher wants to troubleshoot, it is interesting to notice that he does not have the program correct model to reason about, once there is a huge number of different solutions for a single computer problem. Instead, the teacher reasons about program fragment

* This work was supported by an IBM Eclipse Innovation Grant and CAPES.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 7–16, 2006.

models with the semantics of each expression and sentence being given by the programming language. Further, when the teacher tries to really understand the student program intentions, he also reasons in terms of general and well-known programming strategies, trying to identify them into the student code and to check how well they were instantiated to solve the current problem.

An original application of Model Based Diagnosis techniques for program debugging was proposed by Franz Wotawa and Markus Stumptner in an Intelligent Debugging System of the Project Jade, to help advanced programmers to find bugs [2]. In the program model, language expressions and sentences are represented as components; the information flow is represented as a connection; and the components behavior are described based on the language semantics. The diagnostic solution is a set of possible faulty components, in this case, expressions or sentences of the program code with bugs.

Although the Jade Project [3] has achieved some interesting results, we believe that it can not be successfully applied to students. While in the Jade Project it is expected that a programmer will be able to easily correct his bugs by simple looking at the most probably failing parts of his program, a student will find this task very difficult and will probably not be able to learn from that. A step forward to promote such learning process, is to explore the phase where the students makes his predictions about the variable values to be done, in a more comprehensive way. This step goes in the same direction of a teacher trying to understand the student intentions. This can be done by, the use of knowledge about elementary programming strategies, also called *elementary patterns*, to establish a better communication between the diagnosis system and the student.

In this paper we describe the development of an automatic debugging system, named ProPAT_deBUG, which is part of a programming learning environment based on *elementary patterns*, called ProPAT [4]. We also show the results of some experiments realized on a set of student's programs, including faults classified according to a taxonomy of typical student faults.

The rest of this paper is organized as follow. Section 2 gives the background knowledge on Model Based Diagnosis. In Section 3 we discuss how the general theory of MBD can be applied to diagnose student's programs.

2 Model Based Diagnosis Task

2.1 MBD: The Conceptual Model

Diagnosis reasoning can be conceived as performing three subtasks: (1) *symptoms detection*, where a symptom is defined as an observation that deviates from expectation;(2) *hypotheses generation*, where the possible causes, taking into account the initial observation, are generated; and (3) *hypotheses discrimination*, that discriminates the hypotheses set based on additional observations [1].

- **Symptoms Detection.** Symptoms are defined as abnormal observations of the system outputs, i.e, an output value that deviates from the expected value. If no symptoms is detected, the diagnosis task will not be performed.

- **Hypotheses Generation.** The hypotheses generation has two subtasks: *find contributors* and *transform contributors to a hypotheses set.*
  - **Find Contributors.** The contributor set (for the abnormality observations) is a set of components which contains at least one component incorrect (or faulty). The contributor set is also called a *conflict set* [5]. A possible approach to perform this task is by simulation. Simulation yields an expectation (e) for an initial observation (Oinit). If the observation is abnormal, and a set of contributors (c) has been used for deriving the expectation (e), then we say that (c) is the contributor set of the initial abnormal observation (Oinit,ab).
  - **Transform Contributors to Hypotheses Set.** The conflict sets are transformed into a hypotheses set. Every hypothesis in the hypotheses set is an explanation of all initial observations [1] and all its elements are supposed to be faulty. This task can be implemented using the Reiter's Algorithm [6], that constructs the hitting set tree (Section 2.3).
- **Hypotheses Discrimination.** For each hypothesis generated, a set of new observations must be made that will be given as the new input to the hypotheses generation task. To decide in which order the hypotheses are going to be tested, it is necessary to have a fault estimate value derived for each component.

2.2 MBD: The Formal Model

R. Reiter [6] formalized the theory of diagnosis from first principles as follows.

Definition 1. *[6]: A system is defined to be a pair (SD,COMP), where SD is the system description and COMP is a finite set of constants denoting the collection of components of the system. The system description SD is comprised of a set of first-order logic sentences describing the functionality of the components within the system (*behavioral model*) and the connections between the components of the system (*structural model*).*

Definition 2. *[6]: Given an observation, OBS, (SD,COMP,OBS) is a diagnosis problem for the system (SD, COMP) with observation OBS.*

Definition 3. *[6]: A diagnosis Δ for the system (SD, COMP) is a minimal subset of COMP such that:*

$$SD \cup OBS \cup \{AB(C)|C \in \Delta\} \cup \{\neg AB(C)|C \in COMP \backslash \Delta\}$$

is consistent. Where $AB(C)$ means that the component C has an abnormal behavior, i.e., it is faulty.

Definition 4. *[6]: A* contributors set *for (SD, COMP, OBS) is a set $CO \subseteq COMP$ such that $SD \cup OBS \cup \{\neg AB(C)|C \in CO\}$ is inconsistent.*

Definition 5. *[6]: A hitting set for a collection of sets C is a set*

$$H \subseteq \bigcup_{S \in C} S$$

such that $\forall S \in C, H \cap S \neq \emptyset$, i.e., a hitting set is a set that intercepts all the sets of the collection C. A minimal hitting set is an hitting set such that none of its subsets is an hitting set. When the collection set C corresponds to the set of all contributors sets CO for a diagnosis problem (SD, COMP, OBS), the minimal hitting set is the simplest explanation for the observations. The next theorem shows a constructivist form to find the hypotheses from the contributor collection set.

Theorem 1. *[6]:* The set $\Delta \subseteq COMP$ *is a diagnosis for (SD,COMP,OBS) if and only if* Δ is an minimal hitting set for the collection of all contributor sets for the diagnosis problem (SD,COMP,OBS).

2.3 A Brief Review of Reiter's Original Algorithm

Reiter [6] proposed a diagnosis algorithm (for faulty systems in general) that computes all minimal hitting sets for a family of components sets F. The algorithm generates an acyclic graph in which nodes are labeled by sets and arcs are labeled by elements of the set. The idea is that for each node labeled by a set S, the arcs leaving from it are labeled by the elements of S. Let $H(n)$ denote the set formed by the labels of the path going from the root to node n. Node n has to be labeled by a set S such that $S \cap H(n) = \emptyset$. If no such set can be found, the node is labeled by @. The idea is that every path finishing at a node labeled by @ is a hitting set, since it intersects all possible labels for the nodes. The algorithm tries to generate as few new node labels as possible. This is due to the fact that for the diagnosis, the collection of sets F, which can be used as a node labels, will be given only implicitly. Calculating one element of F involves a call to a theorem prover to find a conflict set. The corrected Reiter's algorithm that expands the graph breadth first is:

1. Choose one set of F to label the root node (level 0)
2. For each node n at level i do:

   (a) If n is labeled by a set S, then for every $s \in S$ create an arc departing from n with label s.
   (b) Set H(n) to be the set of arc labels on the path from the root to node n.
   (c) If there is some node n' such that $H(n') = H(n) \cup \{s\}$, then let the s-arc of n point to n'.
   (d) Else, if there is a node n' labeled by @ such that $H(n') \subset (H(n) \cup \{s\})$ then close the s-arc.
   (e) Else , if there is some node n' labeled by S' such that $S' \cap (H(n) \cup \{s\}) = \emptyset$, then let the s-arc of n point to a new node labeled by S'.

(f) Otherwise, let the s-arc point to a new node m and let m be labeled by the first element S' of F such that $S' \cap H(m) = \emptyset$. If no such set exists, then label m by @
(g) If there is some node n' labeled by a set S^* such that $S' \subset S^*$ then relabel node n' by S' and remove all arcs departing from n' which were labeled by elements of $S^* \backslash S'$.

3. Repeat step 2 for level i+1

Theorem 2. *[6]: Let F be a collection of sets and let D be a graph returned by the algorithm above. The set H(n) / n is a node of D labeled by @ is the collection of minimal hitting sets for F.*

3 Model Based Diagnosis of Computer Programs

The basic idea for diagnosing programs, instead of diagnosing physical devices, is to derive a System Description directly from the student program and the programming language semantics. This model must represent components, connections and its behavior based on the actual student program behavior which reflects its errors. The observations are the incorrect outputs in different points of the original program code. The predictions are not made by the system, but by the student and therefore this is the situation where he must communicate his expected values for the variables and be able to underhand what are the possible faults in his program.

There are two approaches that can be used for program modeling: a *value-based model* [2] and a *dependency based model* [7]. They are used to detect functional faults but they are not recommended for detecting structural faults [7]. *Functional faults* are all faults that result from the storage of an incorrect value of some variable, in at least one possible evaluation trace. In particular, these faults include the use of an incorrect operator or the use of incorrect literals (variables). Examples of functional bugs are: omitting an operator (e.g., writing i instead of i+1); using the wrong operator (e.g., writing i++ instead of ++i) or the wrong variable (e.g., a[i] instead of a[j]); missing an initializing of variables; a wrong modification of a value stored in a variable; errors with loop index initializations; exit tests that lead to an erroneous value of a variable. *Structural faults*, on the other hand, are source code bugs, which alter the structure of the underlying program. For example: missing statements, statements out of order, superfluous statements or access to an incorrect variable. As we show in Section 4 the diagnosis system ProPAT_deBUG can detect functional faults and some particular structural faults.

3.1 The Value-Based Model

In the ProPAT_deBUG system we have chosen to use the value-based approach [2] to construct the program model SD. Since the value-based model can eliminate wrong diagnosis by using additional run-time information (the expected values of variables provided by the students), it achieves better

results than the dependency based model in most cases [3] and helps the student to participate on the diagnosis process.

In the value-based model: expressions and statements are represented as components (structural model); the semantics of the expressions and statements are described by sets of logical sentences (behavioral model). Components are connected if there is a flow of information between the corresponding expressions and statements. An information flow between an assignment and another statement occurs, p.e., if the assignment changes the value of a variable that is accessed by other statement, and there is no assignment changing the same variable in between [2]. Thus, to obtain the structural model we make the following:

- all variables are mapped to connections and whenever a variable occurs in an expression, this connection is used to connect the corresponding components. Each time a variable is used in the assignment's left side, a new connection is created and used for all components that use it until the variable is used again in a assignment's left side;
- all sentences, assignments, conditionals, while loops, return statements, method calls and expressions, are mapped into components.

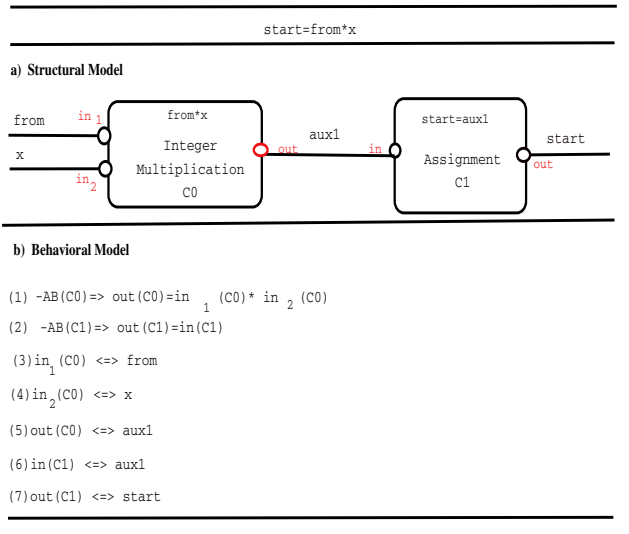


Fig. 1. Simplified structural and behavioral models of the program sentence: start = from * x. The sentence is modeled by two fragments: the IntegerMultiplication (C0) and the Assignment (C1).

The example of structural model on Figure 1a) shows that an assignment component has two ports: input and output port. The input port is related to the evaluation of the expression on the right side of the assignment statement and the output port is related to the variable on the left side of the assignment. A binary operator, for instance, the multiplication operator is mapped to a

component with two inputs and one output port; the output port is related to the evaluation of the relation between the two operands induced by the operator. In order to obtain a complete system description that can be used by the diagnostic algorithm, the behavior of the individual fragments must to be specified. Since each fragment corresponds to a single construction of the programming language, the behavioral model can be derived from the language specification. Figure 1b) shows the behavioral model for the IntegerMultiplication component C0 and the Assignment component C1, where in1(C0) and in2(C0) represent the two inputs of the component C0 and out(C0) is the output of the component C0. The first sentence specify that if the behavior of component C0 is normal ($\neg AB(C0)$) implies that the out(C0) is equal to in1(C0)*in2(C0). In the second sentence in(C1) is the input of C1 and out(C1) is the output. This sentence specifies that if a component C1 is working normally ($\neg AB(C1)$) , out(C1) = in(C1). The sentences 3 to 7 specify the correspondence between the connections and the actual variables of the program code.

3.2 ProPAT_deBUG System

The ProPAT_deBUG diagnosis system analysis the student program, after it has been compiled successfully. To derive the component/connection model from the program, we built a parser in ANTLR [8], a framework for constructing recognizers, compilers, and translators from grammatical descriptions containing Java, C, C++, or Python actions. The diagnosis sub-tasks defined in the conceptual model (Section 2) were implemented in ProPAT_deBUG as follow:

- **Symptom Detection of Programs.** Given a set of test-cases data, represented, for example, as a table of correct input/output values for the problem solution, a symptom is any difference between the program outputs and the outputs of the test-cases. If no symptom is detected, the diagnosis task will not be performed.
- **Hypotheses Generation/Find Contributors.** We implemented this task as a production rule system with a record of dependencies. The rules were constructed such that if a subset of input ad output values are known, new values are computed and propagated. The simulation (constraint propagation) is done by disabling the behavior of the abnormal component (i.e., no values are propagated by it). This allows us to locate faults in expressions, such as wrong operators. Then, to determine the set of contributors in an inconsistency, a strategy of type ATMS was implemented: while the variable values are propagated through the production rules, we keep track of the components that had derived that value and which input values had been used (justifications). After all the values have been propagated, starting from the last contradiction, a backward search finds all constraints that helped to derive the contradiction [5] which will serve as the input for the Reiter's algorithm.
- **Hypotheses Generation/Transform Contributors to Hypothesis Set.** The conflict sets are transformed into a hypothesis set using the Reiter's Algorithm [6], i.e., constructing the hitting set tree, described in Section 2.3.

– **Hypotheses Discrimination.** The fault estimate value is derived from test case execution as follow [9]. To further guide diagnosis, before we start debugging, we first run all test cases and record which fragments are executed for each test cases. As every test case is either classified correct or faulty, we obtain estimate values for each fragment, representing their likelihood to be faulty. After having the hypothesis set ordered by the estimate values, they can then be communicated to the student in that order by asking him about the expected values for the outputs of the components involved in the hypothesis. If the hypotheses bellows to an identified pattern, the system uses a script based on the pattern documentation to establish a better dialogue with the student about his intentions.

4 Empirical Results

In order to evaluate the diagnosis system, we have selected a set of programs with typical faults, classified according with its consequences and its type [7].

The goal of the evaluation of diagnosis system is to measure the ability of the system to return the hypotheses set that includes the real program faults without returning too many plausible hypotheses (i.e., hypotheses that can explain the observations but that are not the real faults). For our goals, since the precision and recall measures are usually defined for the information retrieval area, the sets used in these measures has been redefined for the evaluation diagnosis systems as follow:

FP={set of faults in the student program}

HV={set of the k first hypotheses returned by the diagnosis system ordered by a hypotheses estimate value}

Precision (PRE) : is the ratio of the number of correct hypotheses retrieved by the system ($|FP \cap HV|$) divided by the total number of hypotheses retrieved by the system.

$$PRE = \frac{|FP \cap HV|}{|HV|} \tag{1}$$

By definition, PRE is a measure between 0 and 1: $PRE = 1$, means that all the hypotheses given by the diagnosis system are real faults: $PRE = 0$, indicates that the diagnosis system were not able to find any real hypotheses.

Recall(REC) : is the number of correct hypotheses retrieved by the system divided by the number of faults of the student program.

$$REC = \frac{|FP \cap HV|}{|FP|} \tag{2}$$

That is, $0 \leq REC \leq 1$: $REC = 1$ means that all real faults were considered as hypotheses by the diagnosis system; $REC = 0$ indicates that the diagnosis system were did not considered any of the real faults as hypotheses. It is clear that what one expect from a diagnosis system is a balance between precition and recall.

F-measure : combines both, precision and recall and is defined by:

$$F - measure = \frac{1}{\frac{\alpha}{PRE} + \frac{1-\alpha}{REC}} \tag{3}$$

Where α, $0 < \alpha < 1$ is used as a weight factor to adjust the importance of the precision for the user. The better the F-measure is, the better the precision and recall are. For instance. if we use $\alpha = 0.5$, we give the same importance to the precision and recall, that is:

$$F - measure = \frac{2 * PRE * REC}{PRE + REC} \tag{4}$$

The PROPAT_DEBUG system was evaluated for 28 programs with different types of faults. The results indicate an essential aspect of the program diagnosis: it returns a small number of hypotheses about the student fault. The precision, recall and F-measure was calculated for the set HV with the k-first hypotheses retrieved by the diagnostic system ordered by the fault estimate value of the hypotheses for $k \in \{1, 2, 3\}$.

Using $k = 3$, we observed that the F-measure has shown a better combination between precision and recall. Because these results are not very close to 1, we analyze the importance between precision and recall in the diagnosis, changing the value of α. The left side of Fig. 2 shows the values of the F-measure for $k = 3$ and $\alpha = 0.3$, i.e, we give less importance to the precision than recall. The results are that the majority of problems (64 %) has measure-F between 0.63 and 1. It is important to notice that the recall in the majority of tested programs is big for a little number of hypotheses (Figure 2, right side).

The values of F-measure used in the evaluation of the system shown that the diagnosis is more difficult for structural logical (12 out of the 28 programs analyzed) faults in the student program, while the results were close to 1 for faults of type functional logical. That is, for this type of faults, the diagnosis system always returned the real fault as a hypotheses with few (1 or 2) more hypotheses different from the real.

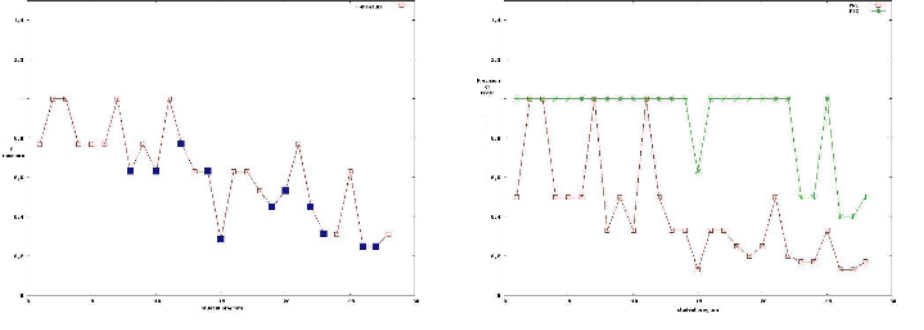


Fig. 2. F-measure, Precision and Recall for k=3 and $\alpha = 0.3$

5 Conclusions and Future Works

We have presented a programming environment, called ProPAT that allows the student to program using elementary patterns, and uses a model based diagnosis system to detect a fault in a student program, named ProPAT_deBUG. The diagnosis system was evaluated with 28 programs with different types of faults according to a classification of typical student faults. The values of F-measure used in the evaluation of the system shown that the diagnosis is more difficult for structural logical faults in the student program. However, the results were close to 1 for faults of type functional logical. That is, for this type of faults, the diagnosis system always returned the real fault as a hypotheses with few (1 or 2) more hypotheses different from the real. The results show that during the interactive debugging process it is possible for a student to learn by answering the questions posed by the AI diagnosis system to discriminate its fault hypotheses about the student program.

Model based diagnosis of programs has not been extensively explored by the automatic debugging community and even less for educational purposes. In fact, the main contribution of this work is to propose an original way of using model based diagnosis to support a programming student learning process.

The most fruitful task for the near future is to run a bigger number of tests and analyze how a group of students, can improve learning using this tool. Another task for the near future is to work more on the elementary patterns' based dialogues. The idea is to extend the number of elementary patterns in ProPAT and to construct natural language templates to communicate the fault hypotheses based on the patterns documentation related to these hypotheses.

References

1. Benjamins, R.: Problem Solving Methods for Diagnosis. PhD thesis, University of Amsterdam (1993)
2. Mateis, C., Stumptner, M., Wotawa, F.: A Value-Based Diagnosis Model for Java Programs. In: Eleventh International Workshop on Principles of Diagnosis (DX). (2000) `http://www.dbai.tuwien.ac.at/staff/wotawa/dx2000c.ps.gz`.
3. Mayer, W., , Stumptner, M., Wieland, D., Wotawa, F.: Observations and Results Gained from the Jade Project. In: International Workshop on Principles of Diagnosis DX-02. (2002)
4. Delgado, K.V., de Barros, L.N.: ProPAT: A Programming ITS Based on Pedagogical Patterns. In: Seventh International Conference on Intelligent Tutoring Systems. Volume 3220., Springer-Verlag Lecture Notes in Computer Science (2004) 812–814
5. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. AI **32**(1) (1987) 97–130
6. Reiter, R.: A theory of diagnosis from first principles. AI **32**(1) (1987) 57–95
7. Wieland, D.: Model-Based Debugging of Java Programs Using Dependencies. PhD thesis, Technische Universität Wien (2001)
8. Parr, T.: ANTLR: Parser generator (1989) `http://www.antlr.org/`.
9. Jones, J., Harrold, M., Stasko, J.: Visualization of test information to assist fault localization (2002)

Intelligent Learning Objects: An Agent Approach to Create Reusable Intelligent Learning Environments with Learning Objects

Ricardo Azambuja Silveira[1], Eduardo Rodrigues Gomes[2], and Rosa Maria Viccari[2]

[1] Universidade Federal de Santa Catarina, Florianópolis – SC, Brazil,
[2] Universidade Federal do Rio Grande do Sul, Porto Alegre - RS –Brazil.
rsilv@ufpel.edu.br, ergomes@inf.ufrgs.br, rosa@inf.ufrgs.br

Abstract. The possibility of reusing learning material is very important to designing learning environments for real-life learning. The reusability of learning material is based on three main features: modularity, discoverability and interoperability. Several researchers on Intelligent Learning Environments have proposed the use of Artificial Intelligence through architectures based on agent societies. Teaching systems based on Multi-Agent architectures make it possible to support the development of more interactive and adaptable systems. We proposed an approach where learning objects are built based on agent architectures. This paper discusses how the ILO approach can be used to improve the interoperability between learning objects and pedagogical agents. It presents the ILO Agent's communication mechanism and a case study.

1 Introduction

This paper addresses the improvement of interoperability among Learning Objects in agent-based Learning Environments by integrating Learning Objects technology and the Multi-Agent Systems approach. A Learning Object, as according to Downes [5], Mohan & Brooks [14] and Sosteric & Hesemeier [15], is an entity of learning content that can be used several times in different courses or in different situations. According to Downes [5], the cost of developing learning materials for e-learning can be large, but as the content of related courses taught at different universities and organizations often tend to be similar, the cost of developing the learning material can be shared among its potential users. The learning object approach promises to reduce significantly the time and the cost required to develop e-learning courses. The use of reusable learning objects to create learning environments improves quickness, flexibility and economy.

A learning object must be modular, discoverable and interoperable, in order to be reused [8]. To achieve these features and improve the efficiency, efficacy and reusability of learning objects, many people have dedicated a great effort. The majority of the focus has been on the definition of standardization. Organizations such as IMS Global Learning Consortium [10], IEEE [9], ARIADNE [2], and CanCore [4], have contributed significantly by defining indexing standards called metadata (data about

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 17 – 26, 2006.

data). Metadata structures contain the information to explain what the learning object is about, how to search, access, and identify it and how to retrieve educational content according to a specific demand.

Mohan & Brooks [14] are among those who point out the limitations of current learning objects. According to them since an instructional designer must carefully examine each learning object, the task of finding the right object may be quite time consuming. In addition, current learning object metadata standards are not very useful in supporting pedagogical decisions.

On the other hand, the state of the art in Intelligent Tutoring Systems (ITS) and Intelligent Learning Environments (ILE) fields points to the use of Agent Society-Based Architectures. MASs have proved to be an appropriate foundation for designing tutoring systems since the teaching-learning problem could be dealt with in a cooperative approach [12]. Using a MAS approach in designing ITS and ILE can result in faster, more versatile and low cost systems. Agents composing such systems are called Pedagogical Agents.

Our research group has been working with these kinds of systems for e-learning [16], [11], [12]. Despite all technological advances that the agents add to our learning systems, our agents are not reusable in various courses and in different situations because they cannot interoperate with other learning environments or agents outside their societies. The use of learning object concepts would enable this feature.

In Silveira et al [17], we proposed the development of learning objects based on agent architectures: the Intelligent Learning Objects approach. The use of agent-based architectures attaches the same features to the learning objects as presented above for agent-based learning environments.

We believe the ILO approach is also useful as a way of enabling the reusability of agents making up pedagogical systems. Hence, this paper will discuss how it can be used to improve interoperability between learning objects and pedagogical agents. The following section reveals some of the basic concepts. Then there is a presentation of the ILO Multi-Agent architecture with its communication structure and a case study showing the use of ILO concepts to improve interoperability among pedagogical agents and learning objects.

2 Intelligent Learning Objects

For this research, an Intelligent Learning Objects (ILO) is an agent enabled to promote learning experiences to students the same way as learning objects do. For this reason, an ILO can be also seen as a learning object built through the agent paradigm. The technological base of this approach is composed by a combination between technologies developed for learning objects and for multi-agent systems.

There are many benefits of integrating learning objects and agents. An Intelligent Agent is a software entity that works in a continuous and autonomous way in a particular environment, generally inhabited by other agents, and able to interfere in that environment, in a flexible and intelligent way, not requiring human intervention or guidance [3]. An agent is able to communicate with others by message exchange using a high-level communication language called Agent Communication Language (ACL), which is based on Logic concepts.

We have started from the learning object model called SCORM [1]. SCORM performs the communication by calling methods (functions) and passing parameters, according to the Object Oriented Programming paradigm. An ILO make uses of ACL for communication among learning objects, hence the learning environments can perform a more powerful communication. With ACL it is possible to supply a more powerful semantic in communication using a formal protocol and a formal Content Language (CL) based on some logic formalism to express the messages content. By using ACL, it is possible to communicate not only variable values, but also facts, rules, mental states and more. The result is that communication by ACL and CL is potentially much better than the communication done by the object oriented approach as the current learning object models do.

Another interesting useful capability of intelligent agents concerns their potential learning ability. This feature gives ILOs the ability to acquire new knowledge and perform different behaviors during its existence, as according to its own experience. Thus, by interaction with students and other ILOs, an ILO is able to evolve. It is not static like current learning objects.

Agents can have coordination and cooperation mechanisms that help the agent society to achieve its goals. Such agent features can be very useful due to the possibility of a self-organizing ILO society where it can promote richer learning experiences. The coordination and cooperation mechanisms enable complex behaviors and interactions among ILOs and, as a consequence, more powerful learning experiences.

Other agent features that may be interesting to promote interaction among learning objects are autonomy, pro-activity, sociability and benevolence. The autonomy of ILO gives it the capability to act based on its own behavior and knowledge with no external intervention. The pro-activity feature assures that the ILO must act in order to satisfy its own objectives. The sociability and benevolence features address the capability of social and cooperative behavior.

2.1 Requirements for Intelligent Learning Objects

The ILO definition proposed in this work permits the construction of many types of ILO with different architecture. However, in order to promote interaction some standardization is needed according to the learning objects and the open agent systems principles.

As a learning object, an Intelligent Learning Objects must be reusable. The reusability is given as a result of three features: interoperability, discoverability and modularity [8]. In Learning Objects approach, the use of metadata to describe the pedagogical content of the learning object gives discoverability. To enable this feature in ILO, we adopted the IEEE 1484.12.1 Standard for Learning Object Metadata [9]. The modularity of learning objects can be reached with a good pedagogical project. So, the design of the pedagogical task of an ILO must be made according to the expertise of some object matter specialists and pedagogical experts. Some interoperability can be reached by the use of well-known standards. For this reason, we adopted two learning object standards: a) the IEEE 1484.12.1 Standard for Learning Object Metadata [9]; and, b) IEEE 1484.11.1 Standard for Learning Technology – Data Model for Content Object Communication. The 1484.11.1 standard is defined for communication of

learning objects with Learning Management Systems (LMS). We use this standard in interactions among ILOs.

For its agent features, ILO adopts some central ideas of intelligent agent principles from Wooldridge et al [18] conceptions. These authors see an agent as coarse-grained computational systems, each making use of significant computational resources that maximizes some global quality measure. Hence, the ILO Agent should not attempt to solve the problem on its own. This is the modularity principle in MAS.

In order to assure interoperability among agents we have adopted the concepts defined by FIPA [7]. The FIPA is an organization which defines standards to enable interoperability for MAS. The main concern of the FIPA is the definition of standards to enable the communication between agents. Having a well-defined communication structure is vital for interoperability among agents. Among FIPA developments are: a language to be used for the communication among agents, the FIPA-ACL; a language to be used for encoding the contents of communication messages, the FIPA-SL; a set of interaction protocols that define patterns of message sequences with associated semantics. We used these technologies to define a communication framework for ILOs. The ILOs must use this framework in order to communicate with each other.

The discoverability in the field of MAS is the ability to be discovered in terms of its tasks and the services it provides. In addition to some services provided by the FIPA architecture, our communication framework contains a set of dialogues that ILOs should use.

Besides those technological issues, one important requirement for an ILO is that it must have an educational purpose. Thus, an ILO must be created and applied in order to carry out some specific tasks to create significant learning experiences by interacting with the student. For this much, the project of an ILO has to involve a contents expert, an educational expert and a computing expert.

3 The ILO Multi-agent Architecture

The agent society presented by Silveira et al [17] encompasses two types of agents: LMS agents and Intelligent Learning Objects. These two kinds of agents are abstractions of Learning Management Systems (LMS) and Learning Objects respectively. Two of the most common entities regarding the learning object technology. However, studies on Silveira's work pointed out the necessity to have a centralized location where agents and users could search for ILOs. Thus, we have introduced the ILOR Agent in the society

Intelligent Learning Objects are agents responsible for playing the role of learning objects. Its responsibility is to generate learning experiences to the students in the same sense of learning objects. LMS Agents are abstractions of Learning Management Systems. It is responsible for dealing with the administrative and pedagogical tasks involving a learning environment as a whole. It provides a way for students to access ILOs, stores information concerning the students, and bestows all student information to the ILO. ILOR Agents are abstractions of Learning Objects Repositories systems. Its responsibility is to store data to permit a user or an agent to find ILOs satisfying a given demand.

Figure 1 illustrates the proposed agent society. Students interact with the LMS Agent in order to gain learning experiences. The LMS Agent searches (with the aid of the ILOR Agent) the appropriate ILO and summons it. The ILO is then responsible for generating learning experiences to the students. In this task it can communicate with the LMS Agent along with other agents in order to promote richer learning experiences. All the communication is performed by messages exchange in FIPA-ACL. The agent environment that these agents inhabit is a FIPA complaint environment. It provides all the necessary mechanisms for message interchanging among the agents.

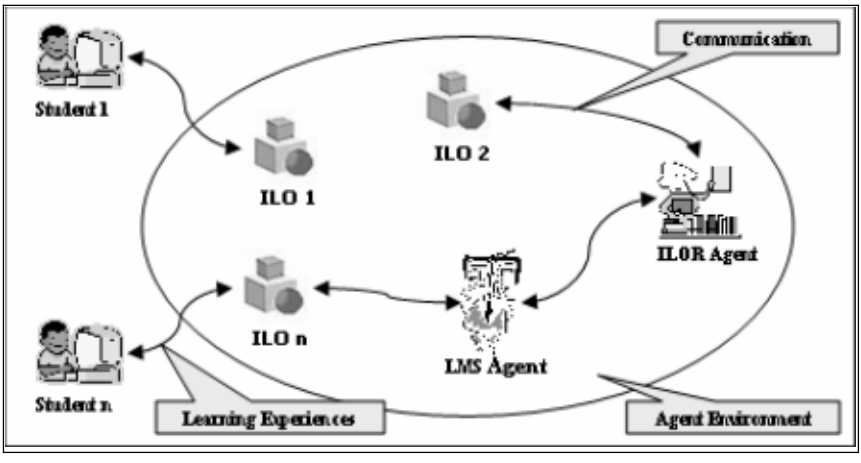


Fig. 1. Proposed agent society

One of the main concerns of this research was the modeling of the communication processes among the agents. Through a well-defined communication framework it is possible to improve interoperability because it enables different types of agents share information with each other. We defined a communication framework based on FIPA reference model [7].

4 Case Study

This section presents a framework composed by a set of Java classes designed to built ILOs as easy as possible. This framework extends the FIPA-OS framework [6], a Java-based toolkit for the development of FIPA compliant agents. The FIPA-OS provides the main services defined by FIPA. A typical FIPA-OS Agent is composed by a main class (sub-class of the FIPAOSAgent class) and some tasks (sub-classes of the Task class). A task is responsible for performing a specific duty the agent is supposed to do.

Using the ILO's communication model the requestor agent acts as the client and the provider agent as the server. Thus, the framework provides services to perform server tasks and client tasks for all the dialogues as shown above. The task classes are ready to be used by the agents. In addition, the framework provides super-classes for

the three types of agents defined in the agent society. These classes perform the main tasks defined for the society's agent, like answering for metadata requests. Figure 3 shows this software architecture. The agent classes extend the FIPAOSAgent class from the FIPA-OS framework. The tasks classes extend the Task class from the FIPA-OS.

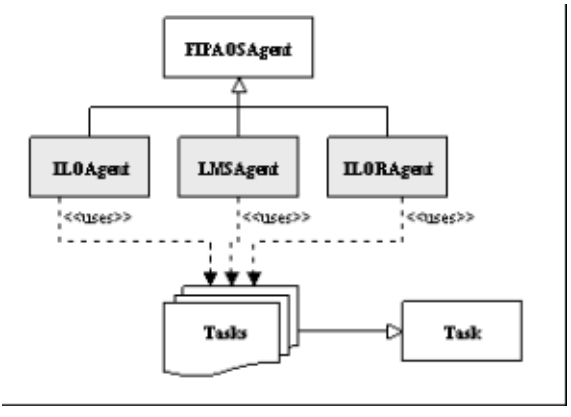


Fig. 2. Framework architecture

The process of building ILOs by using our framework is performed by extending the ILO framework java classes according to the specific learning object features or by adding the necessary tasks to an already implemented FIPA-OS Agent and making modifications to the agent's main class. We use an extended implementation of the agent-based learning environment described in Lucas [13] as a test bed for our approach. The application was used to evaluate the effectiveness of the ILO framework and the proposed approach regarding the interoperability among learning objects, pedagogical agents and agent-based learning management systems

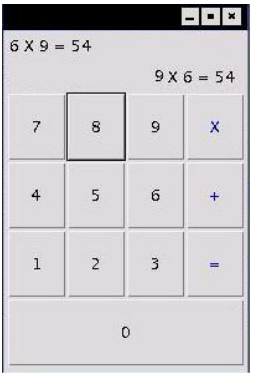


Fig. 3. The Calculator's interface

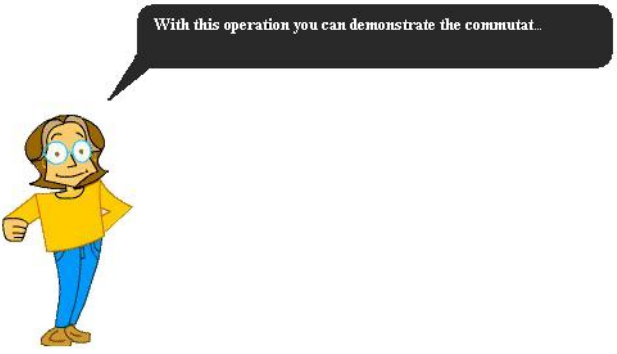


Fig. 4. The APA giving an instruction

The learning environment used as the test bed [13] helps primary school students to learn some fundamental mathematical properties of multiplication (commutative, zero element and identity element) and addition (commutative and identity element). This system is originally composed by a pedagogical agent playing the role of a calculator (see Figure 4) and an Animated Pedagogical Agent (APA) playing the role of an animated tutor (see Figure 5).

The learning environment's life cycle begins when a screen containing the definition of mathematical properties, as well as a tutorial explaining how to apply them, is shown to the student. Secondly, the APA appears on the screen displaying a welcome message and then the calculator is shown on the screen. Thus, the student is able to accomplish the first operation. The APA then informs the student, according to the previously accomplished operation, what mathematical properties can be applied and encourages the student to complete the next operation. After completing the second and last operation, the APA appears again on the screen to inform the student if any mathematical property was applied successfully or not. If the student was not successful on applying some mathematical property, he/she is informed of which were applied incorrectly as well as the mistakes. If at least one property was applied successfully, the APA will just congratulate and inform the properties that were applied successfully. The illustrations below show two elements of the system's interface: the calculator and the APA.

We built the Animated Pedagogical Agent as a LMS Agent and the Calculator Pedagogical Agent in an ILO to get modularity

To transform the Animated Pedagogical Agent into a LMS Agent we added tasks to anable it to act as server for the dialogues get-learner-lms and as put-learner-ilo. We then added the client tasks responsible for registering it in AMS and DF. As this agent was already implemented using FIPA-OS, it only needed to add the tasks we mentioned in the agent's main class and make a few changes in its behavior to get the tasks instantiated when necessary. Now this agent is capable to communicate with other agents about learners using the IEEE 1484.11.1 Standard for Learning Techology – Data Model for Content Object Communication and the dialogues defined in this work.

The first action to transform the Calculator Pedagogical Agent in an Intelligent Learning Object was to describe the pedagogical content of this agent using metadata according to the IEEE 1484.12.1 Standard for Learning Object Metadata and to add the server class of the get-metadata dialogue. With this simple change we enabled the discoverability in this pedagogical agent. We then added the client task of the dialogue put-learner-lms and the server task of the dialogue put-learner-ilo. Now this agent is capable to use the IEEE 1484.11.1 Standard for Learning Technology – Data Model for Content Object Communication in order to communicate with other pedagogical agents. This agent was also implemented using the FIPA-OS. Thus, we just added the mentioned tasks in its main class and changed its behavior a little to get the tasks instantiated when necessary.

The power point presentation had a different approach of adaptation to the ILO approach. It originally was not an agent, so we used the ILOAgent class. We extended this class and added to it the same tasks added to the calculator agent. We also described the agent's pedagogic content using metadata.

The communication between the Animated Pedagogic Agent and the Calculator Pedagogic Agent that was performed in a proprietary cycle was changed to a standardized way using the IEEE 1484.11.1 Standard for Learning Technology – Data Model for Content Object Communication and the put-learner-ilo and put-learner-lms dialogues. All student actions in the calculator's interface are then encoded using the standard IEEE 1484.11.1. These actions are sent to the Animated Pedagogical Agent using the put-learner-lms dialogue. The Animated Pedagogical Agent is responsible for evaluating the student's actions. After this evaluation, the LMS Agent performs its specific duties and requests the calculator once again by using the put-learner-ilo dialogue. If the LMS Agent thinks that it is necessary to show a reading about mathematical properties to the student, it uses the power point presentation by using the dialogue put-learner-ilo. Figure 6 shows this communication structure.

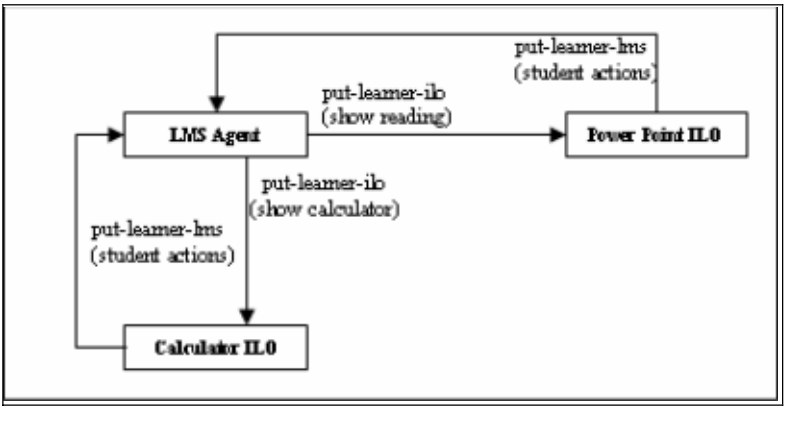


Fig. 5. Communication structure

5 Conclusions

Intelligent Learning Objects are able to improve the adaptability, interoperability and interactivity of learning environments built with these kinds of components by the interaction among the learning objects and between learning objects and other agents in a more robust conception of communication rather than a single method invocation as the object-oriented paradigm use to be.

The test bed application showed that the proposed approach of agent-based learning objects is useful to improve the interoperability of learning objects, pedagogical agents and agent-based learning environments. Besides that, the APA playing the role of a LMS has a real-time interaction with the student. For each action of the student on the calculator, the APA is capable of providing a suitable message to the student and interacts with the learning objects in a powerful way. This interaction provides more reality to the environment, and as a consequence, the student feels that he/she and the APA are active characters of the learning process. In addition, the student is likely to be motivated on learning tasks supplied by the environment, because the human-like character of APA is similar to the student's reality and it have good control on the learning objects.

Acknowledgement

This project is granted by Brazilian agencies: CNPq, CAPES and FAPERGS.

References

1. Advanced Distributed Learning (ADL). Sharable Content Object Reference Model (SCORM ®) 2004 Overview. retrieved July 7, 2005 from www.adlnet.org (2004).
2. ARIADNE Alliance of remote instructional authoring & distribution networks for Europe Retrieved July 7, 2005 http://ariadne.unil.ch (2004).
3. Bradshaw, J. M. An introduction to software agents In: Bradshaw, J. M. Ed. Software Agents. Massachusetts: MIT Press (1997).
4. CanCore - Canadian Core About. Retrieved July 7, 2005 from http://www.cancore.ca/about.html. (2004).
5. Downes, S. Learning objects: resources for distance education worldwide. in International Review of Research in Open and Distance Learning, 2(1) (2001).
6. EMORPHIA. FIPA-OS - FIPA - Open Source. retrieved September 19, 2005 from http://fipa-os.sourceforge.net/index.htm. (2005).
7. FIPA: The foundation for Intelligent Physical Agents. Specifications. retrieved July 7, 2005 from http://www.fipa.org. (2002).
8. Friesen, Norm . What are Educational Objects? Interactive Learning Environments, 3 (9).
9. IEEE Learning Technology Standards Committee (2004). Specifications. retrieved July 7, 2005 from http://ltsc.ieee.org. (2001).
10. IMS Global Learning Consortium. Current specifications. retrieved July 7, 2005 from http://www.imsglobal.org /specifications.cfm. (2004).
11. Jaques, P. A. et. al. An Animated Pedagogical Agent that Interacts Affectively with the Student. AIED 2003, shaping the future of learning through intelligent technologies. Sydney, Australia. (2003).

12. Johnson, W. Lewis; Shaw, Erin. Using agents to overcome deficiencies in web-based courseware. In: World Conference on Artificial Intelligence in Education, AI-ED, 8., 1997. Proceedings… retrieved July 7, 2005 from www.isi.edu/isd/johnson.html. (1997).
13. Lucas, J. P.; Widges, B.; Silveira, R. A. Inserting Animated Pedagogical Agents Inside Distributed Learning Environments by Means of FIPA Specifications. In: Fourth International Joint Conference on Autonomous Agents & Multiagents Systems (AAMAS) – Proceedings of Agent-Based Systems for Human learning Workshop (ABSHL) 2005, Utrecht (2005).
14. Mohan, P.and Brooks, C. Engineering a Future for Web-based Learning Objects. Proceedings of International Conference on Web Engineering, Oviedo, Asturias, Spain (2003).
15. Sosteric, Mike, Hesmeier, Susan. When is a Learning Object not an Object: A first step towards .a theory of learning objects. International Review of Research in Open and Distance Learning ISSN: 1492-3831 (2002).
16. Silveira, R. A.; Vicari, R. M. Developing Distributed Intelligent Learning Environment with JADE - Java Agents for distance Education Framework. Lecture Notes in Computer Science, Biarritz, v. 2363 (2002). 105-118.
17. Silveira, R. A., Gomes, E. R, Vicari, R. M. Inteligent Learning Objects: An Agent-Based Approach of Learning Objects. In Weert, Tom Van, Tatnall, Arthur (Eds.) Information and Communication Technologies and Real-Life Learning.Boston Springer, (2005). 1103 - 110.
18. Wooldridge, M.; Jennings, N. R.; Kinny, D. A methodology for agent-oriented analysis and design. In: Proceedings of International Conference on Autonomous Agent AAMAS, v.3. (1999).

An Experimental Study of Effective Feedback Strategies for Intelligent Tutorial Systems for Foreign Language*

Anita Ferreira

Universidad de Concepcion, Concepcion, Chile
aferreir@udec.cl

Abstract. This paper aims to inform the design of feedback strategies in ITS for Foreign Language. We explore empirical evidence about effectiveness of feedback strategies used in an experimental study in which students interacted with a web-based tutoring program. Results suggest that an ITS for a foreign language should implement feedback which prompts students for answers with grammar errors.

1 Motivation

Most research in ITS has investigated feedback and guidance moves (such as prompting, hinting, scaffolding, and pumping) in teaching procedural skills in domains such as algebra, geometry, physics and computer programming [11,2]. However, little effort has been put into areas such as ITS for foreign languages. The research on tutoring effectiveness has focused on identifying the repertoire of tactics or moves available to tutors [6] such as giving explanations, giving feedback, and scaffolding. In particular, these studies have tried to determine how tutors decide and choose among these different tactics, how they generate explanations and feedback, and what variety of hints they use.

Empirical studies on human tutoring have also been carried out to analyze the tutor responses to error-ridden student contributions. Graesser et al. [3] stated some relevant facts on the complexity of this kind of dialogue move. An effective tutor should give the student feedback in relation to the student's contributions so the tutor can handle the errors by acknowledging that the error occurred, identifying where the error occurred, instructing the student how to repair the error, diagnosing the bugs and misconceptions that generated the error and setting new goals that remediate the error, bugs, and misconceptions,etc. The tutors' feedback moves are responsible for students' learning in a procedural skill. Hume et al. [4] states that the tutors' desire to encourage active learning convinces them to prompt the student with hints. Hinting or reminding is a

* This research is sponsored by the National Council for Scientific and Technological Research (FONDECYT, Chile) under grant number 1040500 "*Effective Corrective-Feedback Strategies in Second Language Teaching with implications for Intelligent Tutorial Systems (ITS) for Foreign Languages (FL).*"

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 27–36, 2006.

strategy that stimulates the recall of inert knowledge or activates the inferences needed in the completion of a task. In trying to put Hume et al.'s theory of hints into practice in the *CIRCSIM* tutor, Zhou and colleagues [11] found the theory to be too broad and too hard to simulate. There is a significant evidence that a great deal of tutors' tactics can be reframed as prompting or encouraging students to construct knowledge, either through the use of content-free prompts or scaffolding prompts [2]. Scaffolding (or scaffolding episode) has been considered to be a pivotal kind of adult-child interchange in which the adult "guides" the child to develop and achieve to the child's fullest potential.

We explore empirical evidence about the type, frequency and effectiveness of feedback strategies based on studies involving three different learning contexts: an observational study of face-to-face classroom interactions, a case study of one-on-one tutorial interactions, and an experimental study in which students interacted with a web-based tutoring program. The results of our empirical studies were similar with regard to the type, frequency and effectiveness of corrective feedback. In this paper, we focus on an our experimental study of feedback strategies implemented in a web-based computer tutor program. We propose here that the incorporation of effective teaching strategies into ITS for Spanish as a foreign language can be informed by the analysis of effective feedback strategies used by students in a a web-based tutoring program.

1.1 Feedback in ITS for Second Language Learning

ITS for FL have incorporated NLP techniques to analyze learners' language production or model their knowledge of a foreign language, in order to provide learners with flexible feedback and guidance in their learning process. These systems use parsing techniques to analyze the student's response and identify errors or missing items. This allows systems, such as those of [10,5,7], to produce sophisticated types of feedback, such as meta-linguistic feedback and error reports, to correct particular student errors. Sams [10] included information about the students' errors as a type of feedback in the BRIDGE ITS, a multimedia tutoring system for German. When using BRIDGE, students receive feedback about the correctness of their responses for all exercise types.

Levin and Evans [5] also argue that ITS for FL can benefit from NLP technology, enabling systems to produce error feedback based on linguistic analysis. They developed the ALICE-chan system which can identify the location of errors and explain the errors in terms of linguistic relations. However, the feedback provided by ALICE-chan is not pedagogically optimal because it uses technical terms which may be confusing to the student.

Although ITS for FL have been developed, there have been few empirical studies demonstrating the effectiveness of feedback in these systems. Nagata [7] investigated the effectiveness of two types of CALL feedback: traditional feedback that indicates only missing or unexpected words in the learner's response, and feedback that provides further information about the nature of the errors in the form of meta-linguistic rules. The results of an achievement test, followed by a retention test three weeks later, showed that the second type of feedback

was more effective than the first for improving the grammatical proficiency of learners of Japanese as L2 in the use of complex structures.

2 The Experimental Study: Analysis of Effective Feedback Strategies

The results of two previous observational and tutorial studies suggest that for grammar, an ITS for a foreign language should implement ways to prompt students' answers using meta-linguistic cues, elicitation and clarification-requests. There is a tendency for Prompting Answer Strategies (PAS) to be more effective than Given Answer Strategies (GAS) for dealing with grammar errors. Indeed, the prompting strategies seem to promote more constructive student learning in a tutorial context [2] because they encourage the student to respond more constructively than when the teacher gives a simple repetition of the answer or a correction of the error. In second language teaching, **Corrective Feedback** is an indication to a learner that his or her use of the target language is incorrect. In our studies, we classified corrective feedback strategies identified in the SLA literature into two groups:

1. **Giving-Answer Strategies (GAS):** Types of feedback moves in which the teacher directly gives the target form corresponding to the error in a student's answer, or shows the location of the student's error. These include:
   (a) **Repetition** of the error or the portion of the learner's phrase containing the error, using stress or rising intonation to focus the student's attention on the problematic part of the utterance. E.g., S: "Future" (Incorrect tense); T: "¿Future?"
   (b) **Explicit correction:** The teacher provides the correct target form. E.g., S: "Cuando ella andó." *(When she went)*; T: "andaba." This differs from recast because the teacher directly corrects the error without rephrasing or reformulating the student's answer.
2. **Prompting-Answer Strategies (PAS):** Types of feedback moves in which the teacher pushes students to notice a language error in their response and to repair the error for themselves. We have called this group prompting answer strategies because of the similarity these strategies bear to the notion of "prompting" described in [2]. This group includes two types of strategies:
   (a) **Meta-linguistic cues:** The teacher provides information or asks questions regarding the correctness of the student's utterance, without explicitly providing the target form. E.g., S: "Compra" (to buy); T: "Tienes que poner un condicional." *(You have to use a conditional.)*
   (b) **Elicitation:** The teacher encourages the student to give the correct form by pausing to allow the student to complete the teacher's utterance, by asking the student to reformulate the utterance, or by asking questions to elicit the correct answer, such as "How do we say that in Spanish?" E.g., T:Dónde está Jorge (Where is Jorge?) S: "Jorge está ..." (Jorge is...); T: "En la..... (In the....)"

By distinguishing the PAS and GAS groups in our study, we hope to gain further insight into the relative merits of feedback strategies that encourage students to attempt to generate or construct the correct form themselves (PAS), versus those in which the teacher resolves the language error either by indicating the location of the error or providing the target form (GAS). This distinction is motivated by Chi et al.'s [2] study arguing for the benefits of constructive learning, in which the student as an active learner constructs an understanding by interpreting new material in the context of prior knowledge by, for example, making inferences, elaborating the material, by integrating material, and so forth. Knowledge construction may occur as a result of self-explaining (either spontaneously or as the result of elicitation), asking questions, responding to the teacher's questions, etc. As in many previous empirical studies in the ITS literature, Chi et al. [2] studied one-on-one human tutorial interactions, in an attempt to identify the features that make tutoring such an effective learning intervention. Their studies suggest that one-on-one tutoring is effective because it provides students with opportunities to engage in constructive activities, and they argue that ITSs should implement ways to elicit constructive responses from students.

To design the materials and procedures for this experimental study, we developed a teaching component concerning aspects of the subjunctive mood that would help learners to improve their grammatical skills in various ways. This component considered PAS (elicitation and meta-linguistic cues) and GAS (repetition of error and correction) for corrective feedback. For correct answers, positive acknowledgments were considered. The present study addresses the research question: *Are PAS or GAS feedback strategies more effective for teaching the Spanish subjunctive mood for foreign language learners?* by addressing the following hypotheses:

- *Hypothesis 1:* Learners who receive PAS after their subjunctive errors will show greater ability to produce this mood correctly, as measured by pre-test post-test gain scores, than learners not exposed to this feedback.
- *Hypothesis 2:* Learners who receive GAS after their subjunctive errors will show greater ability to produce this mood correctly, as measured by pre-test post-test gain scores, than learners not exposed to this feedback.
- *Hypothesis 3:* Learners who receive PAS after their subjunctive errors will show greater ability to produce this mood correctly, as measured by pre-test post-test gain scores, than learners who receive GAS after their errors with this mood.

2.1 Methodology

Participants. Two groups of students participated in the experimental study. The first group was composed of 6 adult students from a Scottish College. They were all enrolled in the first term of an advanced Spanish course. There was 1 male and 5 females, ranging in ages from 25 to 50 years old (average=34.8).

Participants in the second group were 18 young Jamaican university students. All were enrolled in the second year of a high-intermediate Spanish as a FL course. There were 4 males and 14 females, ranging in age from 18 to 21 years old (average=19.5).

The vast majority of the participants reported English as their first language. The exceptions were two students from the Scottish College, whose L1 languages were French and Portuguese. It is important to note that students were not paid for their participation in the study.

2.2 Design of the Experiment for Spanish Subjunctives

In order to determine the effectiveness of feedback strategies in the context of Spanish as a FL, it was necessary to employ a pre-test post-test control group design. Participants were randomly assigned to form three groups, each one containing eight students:

- **The PAS group:** After the pre-test, the participants received PAS feedback for dealing with the incorrect answers and positive acknowledgements for correct answers during the three treatment sessions, and then the post-test.
- **The GAS group:** After the pre-test, the participants received GAS feedback for dealing with the incorrect answers and positive acknowledgements for correct answers during the three treatment sessions, and then the post-test.
- **Control group:** After the pre-test, the participants received only positive and negative acknowledgements after their answers during the three treatment sessions, and then the post-test.

The students did not receive instruction on the subjunctive mood immediately before the experiment. All students did same sessions and worked with the same material, the only difference was about feedback strategies that they received (PAS, GAS or acknowledgment). The instructional tasks were designed to elicit planned writing production and the recognition of two aspects of the subjunctive mood. The students were invited by email to participate in a series of activities to practise their Spanish and get useful tourist information about Chile.

3 Web-Based Computer Tutor Interface

Each test and activity was constituted by ten exercises, eight about subjunctive and two about indicative mood. These last were included in order to keep the participants on their toes. In order to carry out the experiment for the different activities, a simple web interface was designed to allow students to do the tests asynchronously, that is, it is available to the student anytime, anywhere. The interaction begins with a starting page giving instructions about the experiment as a whole and a personal data entry form. As a student first enters his/her personal details, these are registered and the student is automatically assigned to one of the three experimental groups: GAS, PAS or Control. In order to keep

a balanced sample as much as possible, students are assigned to groups in the following way. The first one receives PAS, the second one receives GAS, the third receives Control feedback, the fourth one receives PAS, and so on.Next, the student starts answering the 10-question pre-test in which the first 5 are fill-in-the-blank questions, and the rest are multiple-choice questions.Since the students finish the activities at different rates, an internal state register is enabled to allow the students to move on at their own rhythm. Once a student starts answering a question, the feedback is provided depending on the type of error that occurred.

4 Analysis and Results

The experiment entailed an independent variable and a dependent variable. The independent variable was the group: (1) the PAS group; (2) the GAS group; (3) the control group. The dependent variable was the difference scores that participants earned between their pre-test and post-test scores. With regard to our first hypothesis *"Learners who receive PAS after their subjunctive errors will show a greater ability to produce this mood correctly, as measured by pre-test post-test gain scores, than learners not exposed to this feedback"*, Tables 1 and 2 show the different scores between the pre-test and post-test for the PAS and control groups. As can be seen, the progress made by the PAS group was much more substantial. This suggests that the participants showed steady improvement in accuracy on the subjunctive mood in the syntactic frames involved.

Table 1. Pre-test and Post-test Results of the PAS Group

	Advanced		High-intermediate							
Test	S1	S2	S3	S4	S5	S6	S7	S8	Total	$\bar{Sc}$
Pre-test	1	2	3	3	5	5	7	4	$\frac{30}{80}(38\%)$	3.8
Post-test	9	8	5	7	6	8	8	7	$\frac{58}{80}(73\%)$	7.3
Difference	8	6	2	4	1	3	1	3	28	3.5

For the purpose of statistically measuring gain in learning, we calculated the differences between the average scores of the post-test and that of the pre-test for each group (Tables 1 and 2), which indicates either an actual gain (if the difference is positive), or a loss (if the difference is negative). The analysis shows that gain scores between the pre-test and post-test for the participants who received PAS feedback were statistically more reliable (average score ($\bar{Sc}$)=3.5; $t - test = 4.04, df = 7, p < 0.005$) than those of control group participants ($\bar{Sc}$=0.7; $t - test = 3, df = 7, p < 0.02$) as predicted by hypothesis 1. In addition, the differences between PAS-gain and control-gain ($t - test = 3.422, df = 7, p < 0.02$) were slightly significant.

Concerning our second hypothesis, *"Learners who receive GAS after their subjunctive errors will show a greater ability to produce this mood correctly,*

Table 2. Pre-test and Post-test Results of the Control Group

	Adv.		High-intermediate							
Test	S17	S18	S19	S20	S21	S22	S23	S24	Total	$\bar{Sc}$
Pre-test	1	5	7	3	3	5	5	4	$\frac{33}{80}(41\%)$	4.1
Post-test	2	5	7	3	4	7	6	5	$\frac{39}{80}(49\%)$	4.9
Difference	1	0	0	0	1	2	1	1	6	0.6

as measured by pre-test-post-test gain scores, than learners not exposed to this feedback", the results in Table 2 and 3 suggest that the progress made by the GAS group was better than the control group. However, gain scores between the pre-test and post-test for the participants who received GAS feedback were significantly less reliable ($\bar{Sc}$=1.9; $t-test = 2.53, df = 7, p < 0.04$) than those of the control group ($\bar{Sc}$=0.7; $t-test = 3, df = 7, p < 0.02$). In addition, the differences between GAS-gain and control-gain scores were weakly significant ($t-test = 2.312, df = 7, p < 0.06$).

Table 3. Pre-test and Post-test Results of the GAS Group

	Adv.		High-intermediate							
Test	S9	S10	S11	S12	S13	S14	S15	S16	Total	$\bar{Sc}$
Pre-test	2	5	7	3	6	5	6	3	$\frac{37}{80}(46\%)$	4.6
Post-test	8	5	7	4	7	9	7	5	$\frac{52}{80}(65\%)$	6.5
Difference	6	0	0	1	1	4	1	2	15	1.9

In accordance with hypothesis 3, *"Learners who receive PAS after their subjunctive errors will show greater ability to produce this mood correctly, as measured by pre-test-post-test gain scores, than learners who receive GAS after their errors with this mood"*, Tables 1 and 3 show the difference scores between pre-test and post-test for the PAS and GAS groups. As can be seen, the progress made by the PAS group was better than the GAS group. This suggests that participants who received PAS after their subjunctive errors showed an improvement in accuracy with this mood.

Gain scores between the pre-test and post-test for the participants who received PAS feedback were statistically more reliable ($\bar{Sc}$=3.5; $t-test = 4.04, df = 7, p < 0.005$) than those who received GAS feedback ($\bar{Sc}$=1.9; $t-test = 2.53, df = 7, p < 0.04$) as predicted by hypothesis 3. In addition, considering the small numbers of data, the differences of PAS-gain with GAS-gain were suggested to be somewhat significant ($t-test = 2.56, df = 7, p < 0.04$).

5 Implications for the Design of ITS for FL

The implementation of strategies in our PAS group requires that the ITS for FL be able to carry on an appropriate interaction with the student. Although

unconstrained conversation of the type that human teachers employ is beyond reach, recent advances in tutorial dialogue systems research make the interactive techniques we propose more feasible than they were at the time many ITS for FL were designed. This research shows that sophisticated interactions can be carried on in domains for which rich underlying models have been developed [8,12], or for which possible correct and incorrect responses have been enumerated and feedback moves for each case have been authored [3,9]. In addition, recent work has shown that an ITS in which students produced self explanations by making selections from a menu led to learning outcomes that were equivalent to a version of the system in which students explained their problem-solving steps in their own words [1]. This result suggests that full-blown natural language understanding may not be required in order to support interactions that evoke knowledge construction.

Based of the results of this study and two previous. We have defined a model for the design of a feedback component for ITS for Spanish as a foreign language (Figure 1), which takes into account:the type of error the learner has made (grammar, vocabulary or pronunciation error), and the learner's level of proficiency (beginner, intermediate, advanced). In our model, we assume that error analysis is performed by an interpreter/analyzer. As noted above, prior ITS for FL have made successful use of parsing technology to identify grammar errors, and recent research on a reading tutor has shown that given good expectations about what the student is trying to say, automatic speech recognition can be used to identify pronunciation errors. In this model, we also assume that the learner's level of proficiency is given.

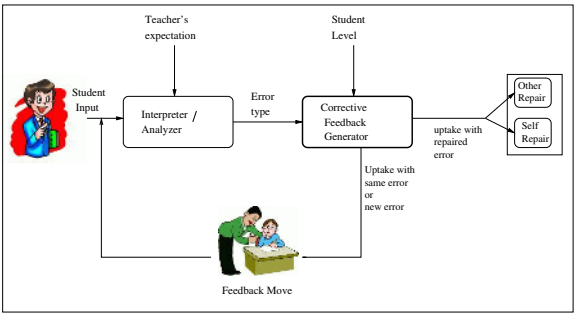


Fig. 1. The Process of Error Treatment and Feedback Generation for an ITS for FL

In our model, the feedback sequence starts when a student's answer contains at least one error. If the answer contains more than one error, the system must determine which error should be treated first, and in our model this decision is based on the learner level. For beginners, grammar and pronunciation errors are the most frequent, and thus we suggest that priority should be given to the treatment of these types of errors. For intermediate and advanced learners, grammar and vocabulary errors should be addressed first. Once an error is identified, a

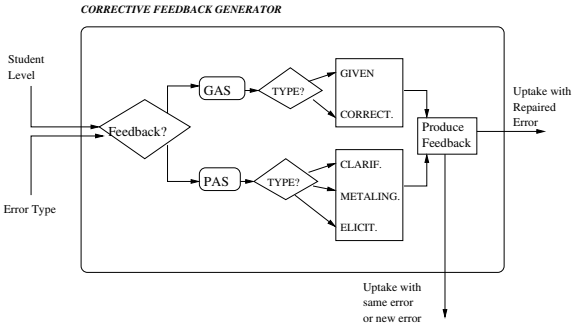


Fig. 2. Model of Corrective Feedback Generation

feedback strategy must be chosen, as shown in the model for feedback generation in Figure 2. After the feedback has been generated, the student may produce several types of responses (uptake):

1. An immediate uptake in which the student modifies his/her answer correctly, either by self-repair (if PAS was generated) or by other-repair (if GAS was generated). This indicates that the student has noticed the error and the given assistance, and the correct answer may indicate a first step towards improvement.
2. An uptake which still contains the error. This may occur because the student did not notice the target form provided by the teacher's feedback or the student does not know how to correct the error. In cases such as this, our human teachers either try an alternative feedback strategy or continue the discussion with the next question, an accept turn, or a domain turn.
3. An uptake in which the student repairs the original error, but his/her answer contains another error. In this case, a feedback strategy is selected according to the algorithm for presenting the first corrective feedback move over an error given above.

A remaining issue that must be addressed in any implementation of our model is how the feedback strategies should be realized in natural language and presented to the student. This will depend on aspects of the overall ITS for FL, such as whether the student interacts with the system using speech or typing (or a combination), whether the ITS for FL is taking a Focus on Forms or Focus on Meaning approach, and so on.

6 Conclusions and Further Issues

This paper addressed the research question of whether feedback strategies (PAS or GAS) are more effective for teaching of the Spanish subjunctive mood for foreign language learners. Overall, the PAS gain was found better than the GAS gain for supporting the process of practicing/learning some aspects of the

subjunctive. After three weeks of the treatment process, learners who received prompting strategies about the sequence of tense and clauses of the subjunctive were significantly more capable of producing the correct forms and of identifying contexts in which the use of subjunctive was appropriate.

Overall, our research approach has been enriched by the research in several disciplines, including Second Language Acquisition, Intelligent Tutoring Systems. These diverse perspectives lead to general questions about how ITS for FL can contribute to alleviating the limitations or disadvantages presented by classroom mode in the treatment of errors, such as giving more opportunity for interaction, prompting student-generated repair. Moreover, the necessity of implementing feedback strategies in ITS for FL can expand our understanding of this key issue and enable us to envisage the kind of contribution that can be useful for ITS for FL systems, as well as teaching training in the context of foreign language instruction.

References

1. V. Aleven, Popescu, and O. Ogan. A formative classroom evaluation of a tutorial dialogue system that supports self-explanation. In V. Aleven, editor, *Procs. of the 11th Int. Conf. on Artificial Intelligence in Education*, pages 345–355, 2003.
2. Michelene T. H. Chi, Stefanie Siler, H. Jeong, T. Yamauchi, and Robert G. Hausmann. Learning from tutoring. *Cognitive Science*, 25(3):471–533, 2001.
3. A.C. Graesser, S. Lu, G.T. Jackson, H. Mitchell, M. Ventura, A. Olney, and M.M. Louwerse. Autotutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, 36:180–193, 2004.
4. G. Hume, J. Michael, A. Rovick, and M. Evens. Hinting as tactic in one-on-one tutoring. *Journal of Learning Sciences*, 5(1):23–47, 1996.
5. L. Levin and D. Evans. ALICE-chan: A case study in ICALL theory and practice. In V. Holland, J. Kaplan, and M. Sams, editors, *Intelligent Language Tutors: Theory shaping technology*. Lawrence Erlbaum Associates, 1995.
6. D. Merrill, B. Reiser, and S. Merrill. Tutoring: Guided Learning by Doing. *Cognition and Instruction*, 13(3):315–372, 1995.
7. N. Nagata. The effectiveness of computer-assisted metalinguistic instruction: A case study in japanese. *Foreign Language Annals*, 30(2):187–199, 1997.
8. E. Owen and K. Schultz. Empirical foundations for intelligent coaching systems. In *Procs. of the Interservice/Industry Training, Simulation and Education Conference*, Orlando, Florida, 2005.
9. C. P. Rosé, P. Jordan, and M. Ringenberg. Interactive conceptual tutoring in atlas-andes. In J. D. Moore, editor, *AI in Education: AI-ED in the wired and wireless future*, pages 256–266. IOS Press, 2001.
10. M. Sams. Advanced Technologies for Language Learning: The Bridge project. In V. Holland, J. Kaplan, and M. Sams, editors, *Intelligent Language Tutors: Theory shaping technology*. Lawrence Erlbaum Associates, 1995.
11. Y. Zhou and R. Freedman. What Should the Tutor do When the Student Cannot Answer a Question? *Procs. of the Twelfth Florida Artificial Intelligence Symposium (FLAIRS'99), Orlando*, 1999.
12. C. Zinn, J. Moore, and M. Core. Intelligent information presentation for tutoring systems. In O. Stock, editor, *Multimodal Intelligent Information Presentation*, volume 27, pages 227–254. Kluwer Academic Publishers, 2005.

Coordination with Collective and Individual Decisions

Paulo Trigo[1], Anders Jonsson[2], and Helder Coelho[3]

[1] Dep. Eng. Elect. Telec. e Comp. at Instituto Superior de Engenharia de Lisboa, Portugal
[2] Departamento de Tecnología at Universidad Pompeu Fabra, Barcelona, Spain
[3] Departamento de Informática at Faculdade de Ciências, Universidade de Lisboa, Portugal
ptrigo@deetc.isel.ipl.pt, anders.jonsson@upf.edu,
hcoelho@di.fc.ul.pt

Abstract. The response to a large-scale disaster, e.g. an earthquake or a terrorist incident, urges for low-cost policies that coordinate sequential decisions of multiple agents. Decisions range from collective (common good) to individual (self-interested) perspectives, intuitively shaping a two-layer decision model. However, current decision theoretic models are either purely collective or purely individual and seek optimal policies. We present a two-layer, collective versus individual (CvI) decision model and explore the tradeoff between cost reduction and loss of optimality while learning coordination skills. Experiments, in a partially observable domain, test our approach for learning a collective policy and results show near-optimal policies that exhibit coordinated behavior.

1 Introduction

A coordination policy recommends decisions expected to bring agents to cooperate on a collective task (e.g. disaster mitigation) or at the very least, not to pursue conflicting strategies (e.g. compete to rescue a victim). The search for a coordination policy that responds to a large-scale disaster, such as an earthquake, is a process beyond individual skills where optimality is non-existent or too expensive to compute [9]. In many cases, communication is insufficient to ensure a single and coherent world perspective. Such communication constraints cause decision-making to occur both at collective (common good) and individual (self-interested) layers, sometimes in a conflicting manner. For instance, an ambulance searches for a policy to rescue a civilian, while the ambulance command center, when faced with a global view of multiple injured civilians, searches for a policy that avoids conflicts and decides which ambulance should rescue which civilian. However, despite the intuition on a two-layered decision, research on multi-agent coordination often proposes a single model that amalgamates those layers and searches for optimality within that model. A centralized model, e.g. the multi-agent Markov decision process (MMDP) [2], builds a purely collective world perspective that is too complex to coordinate and which requires unconstrained communication capability. In a decentralized model, e.g. the multi-agent semi-Markov decision process (MSMDP) [7], an agent makes decisions based on information about the decisions of other agents. If communication does not occur frequently, such information quickly becomes outdated. In addition, the state space of individual agents may become very large, causing learning to be slow. Also,

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 37 – 47, 2006.

there exist game theoretic approaches that require each agent to compute the utility of all combinations of actions executed by all other agents (payoff matrix) which is then used to search for Nash equilibria [10] (where no agent increases his payoff by unilaterally changing his policy); thus, when several equilibrium exist, agents may adhere to individual policies that are not pulled by a collective perspective.

Therefore, our distinctive research hypothesis is: i) a two-layer model intuitively represents the decision-making that occur in complex domains, and ii) a model that includes collective and individual decisions, enables an agent to decide whether a decision should be made at the collective or at the individual level.

We formulate the *collective versus individual*, CvI, two-layer decision model. Each layer is a multilevel decision hierarchy supported on the framework of options [13], which extends the theory of reinforcement learning to include temporally abstract actions. The layers are linked via the formulation of a *collective layer option* over individual options and we use an *inform-request* scheme to communicate between layers; a *regulatory mechanism* provides the means to choose the layer of a decision.

The CvI decision model was experimentally tested in a collectively observable environment [11] (i.e. partially observable, where the combined partial views determines a sole state). The experimental results support our conjectural hypothesis and show how to reduce the learning cost yielding a near-optimal and admissible (from our qualitative analysis) coordination policy.

2 The Framework of Options

A Markov decision process (MDP) is a 4-tuple $M \equiv \langle S, A, P, R \rangle$ model of stochastic sequential decision problems, where S is the set of states, A(s) is the set of admissible actions at state s, R(s, a) is the expected reward when action a is executed at s, and P(s' | s, a) is the probability of being at state s' after executing action a at state s.

The framework of options [13] is founded on the MDP theory and extends the action concept to take variable amounts of time and frames the policy notion into such an extended action. Formally, given an MDP, an option $o \equiv \langle \mathcal{I}, \pi, \beta \rangle$, consists of a set of states, $\mathcal{I} \subseteq S$, from which the option can be initiated, a policy, π, for the choice of actions and a termination condition, β, which, for each state, gives the probability that the option terminates when that state is reached. The computation of optimal value functions and optimal policies, π^*, resorts to the relation between options and actions in a semi-Markov decision process (SMDP). The relation is that "any MDP with a fixed set of options is a SMDP" [13]. Thus, all the SMDP learning methods can be applied to the case where temporally extended options are used in an MDP.

The option representation provides one level of abstraction on top of primitive actions. However, a primitive action, *a*, also corresponds to an option that is available whenever *a* is available, that always terminates after one time step and that selects *a* everywhere. This uniformity defines a multilevel hierarchy in which the policy of an option chooses among other lower-level options. Thus, the agent's decision at each time step is entirely among options, some of which persist for a single time step (one-step option), and others are temporarily extended (multi-step option).

3 The CvI Collective and Individual Layers

The CvI collective layer formulation assumes that agents may be given different option hierarchies (heterogeneous agents), all hierarchies having the same number of levels (depth), i.e. a similar temporal abstraction is used to design all hierarchies.

We first recall (from [12]) the multi-option, $\bar{o}$, formulation: at each time instant the set of agents, Υ, concurrently executes a $|\Upsilon|$-tuple of options, $\bar{o} = \langle o^1, \ldots, o^{|\Upsilon|} \rangle$, such that each agent $j \in \Upsilon$ is executing option $o^j \equiv \langle I^j, \pi^j, \beta^j \rangle$, for j=1,…,$|\Upsilon|$.

In a partially observable environment, an agent may only observe part of the state. Let S^j be the set of partial states observed by agent j, and let H^j be the set of partial states hidden to j. The complete set of states is $S = S^j \times H^j$. For each option available to j, the initiation set is $I^j \subseteq S^j$, which means that the option is available whenever a state $s \in (I^j \times H^j) \subseteq S$, since for agent j the hidden part of the state is irrelevant.

We now formulate (over $\bar{o}$) our *collective layer option*, $o_{\bar{o}}$, concept: at the collective layer, an option has the form $o_{\bar{o}} \equiv \langle I_{\bar{o}}, \pi_{\bar{o}}, \beta_{\bar{o}} \rangle$ such that,

- $I_{\bar{o}} = \cap_{j=1..|\Upsilon|} (I^j \times H^j)$, i.e. $o_{\bar{o}}$ is admissible at any state $s \in S$ whenever each component of $\bar{o}$ is admissible at *s*,
- $\pi_{\bar{o}} = \langle \pi^1, \ldots, \pi^{|\Upsilon|} \rangle$, where π^j is the policy of agent $j \in \Upsilon$ while executing the component o^j of $\bar{o}$. Therefore, $\pi_{\bar{o}}$ (the policy of $o_{\bar{o}}$) is simply to follow the several, $|\Upsilon|$, individual policies,
- $\beta_{\bar{o}} = \tau(\beta^1, \ldots, \beta^{|\Upsilon|})$ where β^j is the termination condition of the component o^j of option $\bar{o}$. Since each o^j component may have a different duration, the τ function represents the termination scheme of the multi-option $\bar{o}$. In our model we always use the $\tau_{continue}$ scheme [12], where $\bar{o}$ terminates as soon as any o^j terminates but without dropping commitments (i.e. non terminating options keep executing); other referenced schemes are τ_{any}, τ_{all} [12] and τ_{change} [1].

The set of agents, Υ, defines an option space, $\bar{O} \subseteq O^1 \times \ldots \times O^{|\Upsilon|}$, where O^j is the set of options specified for agent j and each $o_{\bar{o}} \in \bar{O}$ is a collective layer option. Thus, the number of $o_{\bar{o}}$ options may grow exponentially with the number of agents, e.g. if all agents have the same, O, option space, $|\bar{O}|$ may grow to $|O|^{|\Upsilon|}$. The designer may use domain constraints that reduce $|\bar{O}|$, i.e. prevent some $\bar{o} \in O^1 \times \ldots \times O^{|\Upsilon|}$ from occurring.

We now define the subsets $O^j{}_d \subseteq O^j$ such that all options in $O^j{}_d$ are specified at the same hierarchical level, d (of agent j); similarly we define $\bar{O}_d \subseteq O^1{}_d \times \ldots \times O^{|\Upsilon|}{}_d$ for $0 < d \leq$ *hierarchyDepth*, each constraining $\bar{O}$ to the options available at hierarchical level d (level-0 is the hierarchy root at which there are no options to choose from).

A policy over $\bar{O}_d$ is implicitly defined by the SMDP M_d which is defined over S with the *collective layer options* at level d as options. The M_d solution is the level-d constrained optimal meta-policy, i.e., it is the way to choose, at each state, the level-d individual policies which, in the long run, gather the highest collective reward.

Figure 1 illustrates the two-layer decision model where the individual layer (each agent^i task hierarchy) has 3 levels and thus the collective layer (represented by two, $\bar{o}_1$ and $\bar{o}_2$, possible multi-option instances) contains 2 levels; at each level, the set of diamond ended arcs, links the option $o_{\bar{o}}$ to its current policy $\pi_{\bar{o}}$.

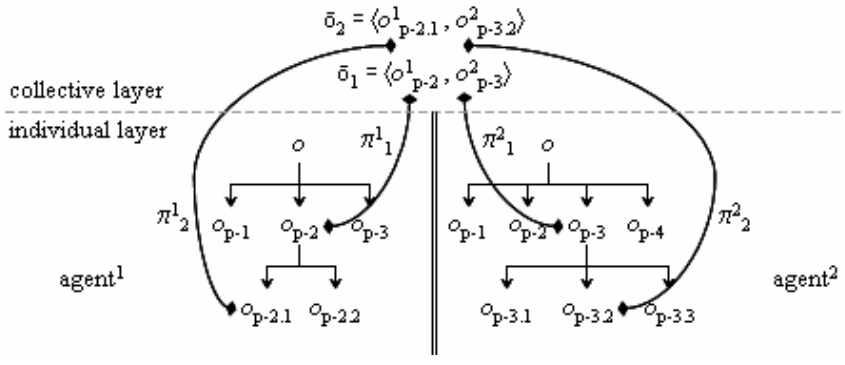


Fig. 1. The two-layer decision model and the links between layers (superscript j refers to agent^j; k subscript refers to the k hierarchical level and p-k subscript refers to the k tree path)

A centralized approach defines the M_d meta-policies and decides which individual policy to follow. Our approach is decentralized and lets each agent decide whether to make a decision by itself or to ask the collective layer for a decision.

In order to make well-informed decisions the collective layer needs to know, once an individual option terminates, about: i) the agent's world observation at that time instant, and ii) the agent's reward acquired since the last reported information. To exchange information we use the *request* and *inform* performatives' FIPA [6] pattern and we specified a simple protocol. At the individual layer: i) an option termination triggers an *inform* to the collective layer containing the agent's local observation and accumulated reward (since its last report), and ii) a *request* contains the option's hierarchal level at which the collective indication is required. At the collective layer: i) an *inform* arrival triggers a *request* for other agents to indicate their execution status, observation and cumulative reward (since last report), and ii) a *request* arrival triggers a decision about which admissible option the requesting agent should follow, *inform*'s the requester and assumes that the requester will follow that option's policy.

4 The CvI Regulatory Mechanism

The first step to instantiate a CvI model is to identify, from each agent's options set, O, the subset of options, $C \subseteq O$, where a meta-policy is most effective to achieve coordination skills; the remaining options, $J = O - C$, are purely individual. A simple, domain-independent design defines C as the set of multi-step options; hence J as the one-step options. Also, the highest hierarchical level(s) are usually effective to achieve coordination skills as they escape from getting lost in the confusion of

lower-level details. Our approach, at its current stage, requires a designer to specify domain-dependent collective and individual options.

The $\mathcal{C}_d \subseteq \mathcal{C}$ and $\mathcal{J}_d \subseteq \mathcal{J}$ sets are used to define each $\bar{O}_d$ option space. At the collective layer, each option, $\bar{o}_d \in \bar{O}_d$, contains all $\mathcal{C}_d$ individual options, plus a special option, indOp_d, that represents $\mathcal{J}_d$ at the collective layer, i.e. indOp_d lets the agent itself choose among individual options. The indOp_d option is always admissible, terminates when the individual layer option being executed terminates and its policy is the agent's policy for choosing among purely individual options.

The second step is to devise the runtime mechanism that regulates the importance the agent credits to individual and collective decisions. The "importance" is a criterion related to a valuation in terms of benefits and costs an agent has of a mental state situation [4]; here, the mental state is the agent's policy space. We materialize the "importance" as the ratio between, the maximum expected benefit, in choosing a collective and an individual option. Thus, we define the *regulatory condition*: "$f(\max_c Q(s, c)) / f(\max_j Q(s, j)) < \kappa$", where $\kappa \geq 0$, $c \in \mathcal{C}_d$ and $j \in \mathcal{J}_d$ and $0 < f(x) = 1/(1+e^{-\chi x}) < 1$. The κ threshold is used to grade the focus from the individual to the collective layer. The f(x) normalizes the action-value function, Q, and the χ parameter configures the significance of "close" values (around the origin).

The usage of the regulatory condition (rC) depends on the kind of agent we want to implement. We used it to design agents that only consider the individual options that are expected to be better than any collective option, i.e. they follow the *behavior rule*: "`if` [rC $\wedge$ ($\max_j Q(s, j) > \max_c Q(s, c)$)] `then` decide-at-individual-layer, `else` *request*-collective-layer". Thus, the *behavior rule* defines: i) if $\kappa = 0$, then "always collective" decision, ii) if $\kappa \geq 1$, "when-possible", i.e. $\max_j Q(s, j) > \max_c Q(s, c)$, individual decision, and iii) if $0 < \kappa < 1$, decision depends on the rC value. The next sections show the cooperative behavior of these agents on different settings.

5 Experiment Specification

We implemented the CvI decision model and tested it in a multi-agent taxi problem (that extends the original single-agent taxi problem [5]), where a maze-like grid is inhabited by taxis (agents), passengers and sites. Passengers appear at a site and wish to be transported to another site. Taxis go to the origin site of a passenger, pick up the passenger, go to its destination site and drop down the passenger. Taxis may pick up several passengers; a site may have several passengers, each with its own destination.

The environment is collectively observable as each taxi does not perceive the other taxis' locations, but their combined observations determine a sole world state. The goal is to learn a coordinated behavior where taxis cooperate to minimize the resources (time) spent to satisfy the passengers' needs.

We made 3 different specifications of individual and collective options: i) *CvI*, defines multi-step options as collective and one-step options as individual, ii) *purely collective*, only considers one-step options, all defined as collective, and iii) *purely individual*, considers that there are no collective options.

The same setup is used for all experiments: 5×5 grid, 4 sites $\mathbb{S}b = \{b1, b2, b3, b4\}$, 2 taxis $\mathbb{S}t = \{t1, t2\}$, and 2 passengers, psg_1 and psg_2. The actions available to each taxi are `pick`, `put`, and `move(`m`)`, $m \in$ {`N`,`E`,`S`,`W`}, the four cardinal directions.

The learning of the collective coordination policy occurs simultaneously with learning of the individual policies and the experiments' results (cf. Section 6) show that, except for the *purely individual*, all other agents exhibit a coordination policy. Also, each agent always learns optimal individual policies, i.e. the best way to execute tasks (e.g. how to navigate to a site and when to pick up a passenger) and their proper order execution (e.g. pick up a passenger at origin before navigating to destination).

Individual Layer Specification. The taxi observation, $\omega = \langle x, y, psg_1, psg_2 \rangle$, represents its own (x,y)-position and status of passenger, $psg_i = \langle loc_i, dest_i \rangle$, where $loc_i \in \mathbb{S}b \cup \mathbb{S}t \cup \{t1_{acc}, t2_{acc}\}$ (j_{acc} means taxi j accomplished delivery) and $dest_i \in \mathbb{S}b$. The rewards provided to a taxi are: i) 20 for delivering a passenger, ii) -10 for illegal `pick` or `put`, and iii) -1 for any other action, including moving into walls.

The same task hierarchy is used both at the *CvI* and at the *purely individual* specifications, which is composed of a `root` option and a `navigate(`b`)` option for each $b \in \mathbb{S}b$ (which, except for the root option is identical to the specification in [8]). The taxi's observation space is denoted by Ω, and $\Omega_b = \{\omega \in \Omega \mid x = x_b \wedge y = y_b\}$ represents the situations where the taxi is at site b. Thus, options are defined as: `navigate(`b`)` $\equiv \langle \mathbb{I}_b, \pi_b, \beta_b \rangle$, where i) $\mathbb{I}_b = \Omega - \Omega_b$, ii) π_b is the policy to learn, and iii) $\beta_b = 1$ if $b \in \Omega_b$ or $\beta_b = 0$ otherwise. The actions `move(`m`)`, $m \in$ {`N`,`E`,`S`,`W`} are the only ones available to π_b policy. The root option of agent j, is `root` $\equiv \langle \mathbb{I}, \pi, \beta \rangle$, where i) $\mathbb{I} = \Omega$, and ii) $\beta = 1$ if $(\exists\ loc_i : loc_i = j_{acc} \wedge \neg\ \exists\ loc_i : loc_i = j)$ or $\beta = 0$ otherwise, i.e. a taxi terminates an episode as soon as it delivers at least one passenger and there are no more passengers in that taxi. The options available to the π policy are: `navigate(`b`)` for each $b \in \mathbb{S}b$, `pick` and `put`.

Therefore, each agent holds an option hierarchy with 3 levels where `root` is the level-zero option, `navigate(`b`)`, `pick` and `put` are the level-one options and `move(`m`)` are the level-two one-step options (defined for each `navigate(`b`)`).

Collective Layer Specification. The collective layer holds the agents' combined observation $\mathbf{s} = \langle ag^1, ag^2, psg_1, psg_2 \rangle$, where ag^j is the (x,y)-position of agent j.

Our approach to the reward is to consider that agents equitably contribute to the current world state. Thus, the collective reward is defined as the sum of rewards provided to each agent; our purpose is to maximize the long run collective reward.

The *CvI* specification considers $\mathcal{C}$ = {`navigate(`b`)` for all $b \in \mathbb{S}b$} and $\mathcal{J}$ = {`pick`, `put`}. These are level-1 options, so $\mathcal{C}_1 = \mathcal{C} \cup \{\mathsf{indOp}_1\}$ and $\mathcal{J}_1 = \mathcal{J}$. The *purely collective* only defines one-step options, each being an option in $\mathcal{C}$. The *purely individual* considers none decision-making at the collective layer, thus defining $\mathcal{C} = \emptyset$.

Within this experimental toy world, an individual agent perceives 25,600 states, and the collective layer contains 640,000 states; a *purely individual* decision considers 6 options, while for *CvI* there are 25 collective options. Hence, the experiments

capture some complexity of a disaster response environment while learning a coordination policy.

6 Experiments and Results

Our experiments were conducted with two main purposes: i) measure the influence of the regulatory mechanism in the coordination learning process, and ii) compare the quality of the achieved coordination, regarding the agents' cooperation behavior.

We ran 4 experiments using the *CvI* specification, each with a different regulatory threshold (cf. Section 0). We ran an additional experiment to compare the *CvI*, *purely collective* and *purely individual* specifications. Each experiment executed 100 episodes. An episode always started in the same state and terminated as soon as all passengers reached their destination.

Policy learning used a temporal difference approach (SMDP Q-learning [13], [3]) with an ε-greedy exploration strategy, which picks a random action with probability ε and behaves greedy otherwise (i.e. picks the action with the highest estimated action value). Each experiment started with $\varepsilon = 0.15$ and ε always decayed 0.004 after each 3 episodes for the last-third of the experiment.

Influence of the Regulatory Mechanism. Figures 2 and 3 show the result of using the regulatory mechanism, throughout the learning process, with the *CvI* specification; each plotted line is labeled `c_`κ after the threshold, κ, used. Figure 2 shows the evolution of the "always collective", `c_0`, and "when-possible individual", `c_1`, (cf. Section 0) learning processes; Fig. 3 uses the collective layer's cumulative reward (at an experiment) and compares the performance for $\kappa \in \{1, 0.7, 0.3, 0\}$.

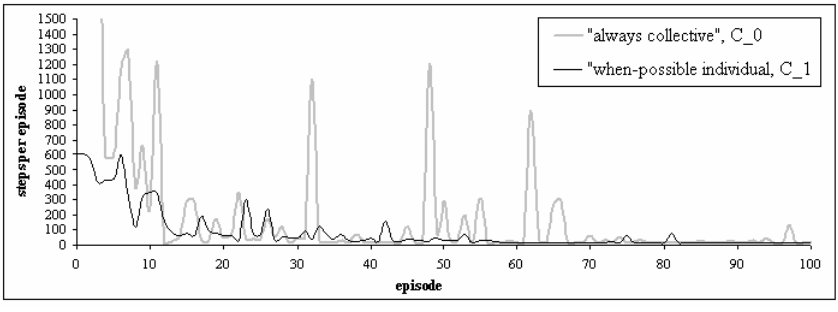


Fig. 2. The "always collective", `c_0`, and "when-possible individual", `c_1`, decisions

Figure 2 shows that `c_0` is a slower and a more instable process than `c_1`. The higher complexity of the collective decision, `c_0`, is revealed mostly during the first and second thirds of the experiment. During the first-third of the experiment, the average number of steps per episode at `c_0` and `c_1` is, respectively 484 and 200; hence, an episode at `c_0` is, on the average, about 2.4 slower than an episode at `c_1`.

The standard deviation is 694 at `c_0` (1.4 times average) and 178 at `c_1` (0.9 times average), which accounts for the instability, that although higher at `c_0`, occurs at both processes. This relative behavior occurs again during the second-third of the experiment, although with lower average and standard deviation values. During the last-third of the experiment both `c_0` and `c_1` show a similar convergent pattern.

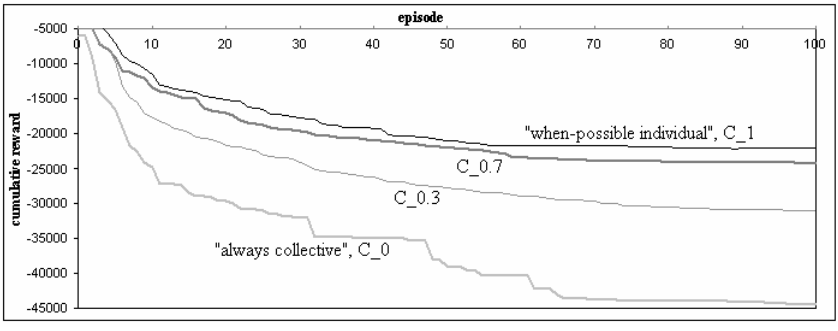


Fig. 3. The influence of the regulatory threshold in the "collective to individual tuning"

Figure 3 confirms that the influence of the regulatory mechanism is most relevant for the first-third of the experiment, where lower values of κ yield a faster decrease of cumulative rewards. It clearly shows that as we move from individual to collective focus (κ decreases) learning a coordination policy becomes more expensive.

All plotted experiences eventually converge to a performance (slope) that, although similar, represents distinct coordination policies (cf. Fig. 5).

An insight on these results is that as κ increases the agent is "left alone" and begins earlier to learn its task; then, when after some episodes the collective options become determinant to solve the task (higher action-values), the opportunity arises for the collective layer to choose options thus "compensating" the initial individual learning.

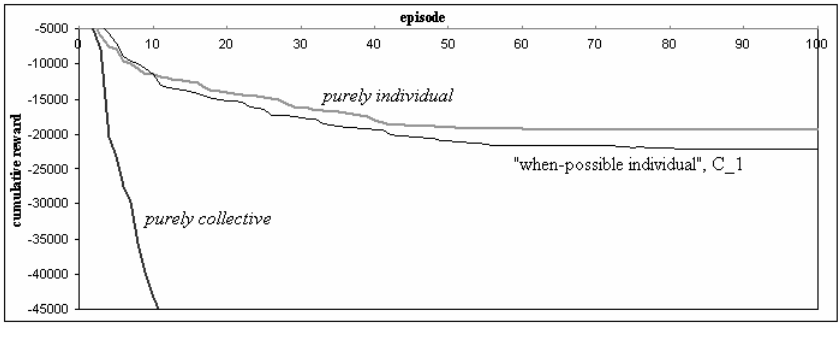


Fig. 4. The relation among specifications: *purely collective*, *purely individual* and *CvI* for κ=1

We recall that, in the "always-collective" case, each agent owns a hierarchical task decomposition and the collective layer may tell an agent to execute an individual option (via indOp) which gives the agent some freedom to decide what to do.

In the *purely collective* case, the collective layer always tells an agent what to do. Figure 4 *purely collective* graph shows a high slope that rapidly decreases rewards, meaning that the collective layer is quickly getting lost in the confusion of low-level details while trying to coordinate the agents (rewards reach ≈ -500,000, at the end of the experiment, still decreasing).

Hierarchical approaches, *purely individual* and `c_1`, exhibit a similar performance with `c_1` being asymptotically better since it leads to higher expected reward; `c_1` coordination policy takes 17 steps when *purely individual* needs 21 steps (cf. Fig. 5).

The Coordination Policies. The performance analysis does not evidence the concrete coordination policy learnt by each process. In order to conclude about coordination we need to analyze the details of a specific episode.

Figure 5 compares the coordination policies achieved at the last episode of `c_0`, `c_0.3`, `c_0.7`, `c_1` and *purely individual*. Each grid shows a different coordination policy, used by the taxis t_1 and t_2, to deliver passengers psg_1 and psg_2 respectively to the s_2 and s_1 sites.

Figure 5 (a) exhibits the optimal coordination strategy, where taxi t_1 although closer to psg_1 decides to pick up psg_2 and deliver at s_1; meanwhile t_2 picks up and delivers psg_1 at s_2. This strategy only takes 16 time steps to terminate the episode because taxi t_1 learnt to follow the cooperative collective perspective.

Figure 5 (b) coordination strategy takes 17 time steps to terminate because t_1 takes the individual perspective and decides to immediately pick up passenger psg_1; taxi t_2 cooperates with t_1's decision as t_2 decides to pick up psg_2 instead of running for psg_1.

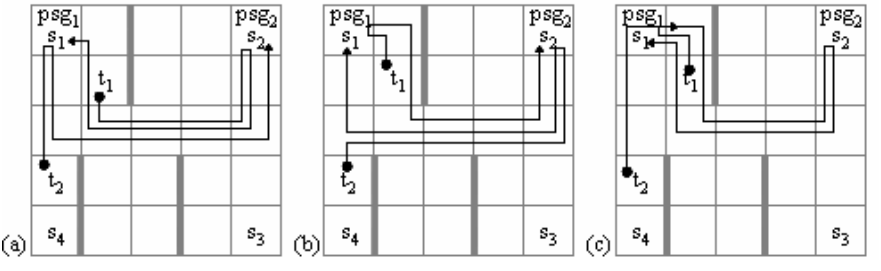


Fig. 5. The coordination policies, achieved by: (a) `c_0` and `c_0.3` with 16 steps, (b) `c_0.7` and `c_1` with 17 steps, and (c) *purely individual* with 21 steps

Figure 5 (c) shows the individual perspective of taxis t_1 and t_2 as they both decide to pick up their closest passenger and they end up competing to pick up the same, psg_1, passenger. Taxi t_1 wins and picks up psg_1 after which they compete again to pick up psg_2. They both arrive at the same time at s_2 and it is interesting to notice that t_1 wins again and picks up psg_2. Taxi t_1 wins because instead of immediately dropping down psg_1 (that t_1 was carrying) it first picks up psg_2 and only then it drops down psg_1.

7 Conclusions

This work explores the separation of concerns between collective and individual decisions while learning coordination policies in a multi-agent complex environment. The separation of concerns frames a two-layer decision model where the definition of inter-layer relations enables the agent to decide at which behavioral layer a decision should be taken. Such two-layered decision-making represents our novel contribution to multi-agent coordination within a reinforcement learning framework.

The formulation of the two-layer, CvI, multilevel hierarchical decision model and the definition of an inter-layer regulatory mechanism enable us to show experimentally how to explore the individual policy space in order to decrease the complexity of learning a coordination policy in a partially observable setting.

The two-layer approach augments the design flexibility in two ways: i) makes it possible to specify individual task hierarchies that are not necessarily equal, therefore allowing for agents' heterogeneity, and ii) enables to configure different architectures (e.g. centralized or decentralized) depending on the information exchange between collective and individual layers.

Future work will explore, at the collective layer, the inclusion of state abstraction [8] and the integration of teamwork strategies [14] to search the most distinguishing state space and to reduce the space of admissible actions, respectively. We aim to apply the resulting technique to a simulated disaster response environment [9].

Acknowledgements. Research was partially supported by PRODEP III 5.3/13/03.

References

[1] Abdallah, S. and Lesser, V. (2005). Modeling Task Allocation Using a Decision Theoretic Model. Fourth Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS05), pp. 719–726. ACM Press.

[2] Boutilier, C. (1999). Sequential Optimality and Coordination in Multi-Agent Systems. Sixteenth Int. Joint Conference on Artificial Intelligence (IJCAI99). pp. 478–485.

[3] Bradtke, S. and Duff, M. (1995). Reinforcement learning methods for continuous time Markov decision problems. Advances in Neural Inf. Processing Systems 8, pp. 393–400.

[4] Corrêa, M. and Coelho, H. (2004) Collective Mental States in Extended Mental States Framework. International Conference on Collective Intentionality.

[5] Dietterich, T. (2000). Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. Artificial Intelligence Research 13, pp. 227–303.

[6] FIPA Communicative Act Library Specification, http://www.fipa.org, 2002.

[7] Ghavamzadeh, M.; Mahadevan, S.; Makar, R. (2006). Hierarchical Multi-Agent Reinforcement Learning. Journal of Autonomous Agents and Multi-Agent Systems.

[8] Jonsson, A. and Barto, A. (2001). Automated State Abstractions for Options Using the U-Tree Algorithm. Advances in Neural Inf. Processing Systems, 13, pp. 1054–1060.

[9] Kitano, H.; Tadokoro, S.; Noda, I.; Matsubara, H.; Takahashi, T.; Shinjou A.; Shimada, S. (1999). RoboCup Rescue: Search and Rescue in Large-Scale Disasters as a Domain for Autonomous Agents Research. Conf. on Man, System and Cyb. (MSC-99), pp. 739–743.

[10] Nash, J. (1951). Non-Cooperative Games. Annals of Mathematics, 54, pp. 286-295.

[11] Pynadath, D. and Tambe, M. (2002). The Communicative Multiagent Team Decision Problem: Analyzing Teamwork Theories and Models. Journal of AI Research, 389–423.
[12] Rohanimanesh, K. and Mahadevan, S. 2003. Learning to Take Concurrent Actions. Sixteenth Annual Conference on Neural Information Processing Systems, pp 1619–1626.
[13] Sutton, R.; Precup, D.; and Singh, S. (1999). Between MDPs and Semi-MDPs: A framework for temporal abstraction in reinforcement learning. A.I., 112(1-2), pp181–211.
[14] Trigo, P. and Coelho, H. (2005). The Multi-Team Formation Precursor of Teamwork. EPIA05, LNAI 3808, pp. 560–571. Springer-Verlag.

Negotiator Agents for the Patrolling Task

Talita Menezes, Patrícia Tedesco, and Geber Ramalho

Universidade Federal de Pernambuco
Centro de Informática, Cx. Postal 7851, 50732-970. Recife – PE, Brasil
{trm, pcart, glr}@cin.ufpe.br

Abstract. Multi-agent systems can be used to perform patrolling tasks in various domains. In this work, we compare the results obtained by new negotiation based approaches with previous ones. By splitting the nodes of the world graph, the negotiator agents reduce the path they have to walk and the number of nodes to patrol, making it easier to maintain a low average idleness in world nodes. Auctions are the negotiation mechanisms used to split the nodes of the world, the agents bid on nodes based in their utility function. Empirical evaluation has shown the effectiveness of this distributed approach, as the results obtained are substantially better than those previously achieved by negotiator agents. The agent types presented in this work are more scalable and reactive since they can perform patrolling in worlds of all sizes and topology types. Besides, they are more stable as indicated by the low standard deviation obtained in node idleness.

1 Introduction

Multi-agent patrolling is a task where agents must coordinately visit some places, at regular intervals, in order to protect or supervise them. Informally, a good patrolling strategy should minimize the time lag between two visits to the same place or to all places. Multi-agent patrolling can be useful for domains where distributed surveillance, inspection or control is required, such as: computer network administration, robotics, and, especially, computer games [1].

Previous works on multi-agent patrolling have been done, exploring different techniques, ranging from machine learning to utility-based agents and operation research. The comparison of these works has shown [2] that none of the proposed solutions is clearly the best with respect to all the criteria for a good patrolling solution. In particular, there were problems of scalability. In this work, we improve a preliminary solution based on negotiation mechanisms, integrating it with advanced decision making and path-finding techniques. We have achieved very good results, especially with respect to scalability, adaptability and stability of the patrolling.

This paper is organized as follows. Section 2 describes the Patrolling task; section 3 presents previous work; the approaches developed in this work are described in section 4; the experiments and results are shown in section 5. Finally, section 6 presents our conclusions and discusses future research.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 48 – 57, 2006.

2 The Patrolling Task

One of the major contributions of [1], the pioneer study on this subject, was the identification of *evaluation criteria* for patrolling. Considering that a cycle is a simulation step, the *instantaneous node idleness* (or simply idleness) for a node *n* at a cycle *t* is the number of cycles elapsed since the last visit before *t* (i.e. the number of cycles the node *n* remained unvisited). The *average idleness* is the mean of the instantaneous graph idleness over a t-cycle simulation. In the same context, another interesting measure is the *worst idleness*: the highest value of the instantaneous node idleness encountered during the whole simulation.

Besides the *evaluation criteria* proposed in [1], there are other important measures that have not yet been considered. The ones we propose in this work are: scalability, stability, adaptability and reactiveness. The scalability criterion evaluates if the approach is capable to patrol worlds of all sizes. Previous works [1-4] have always simulated in graphs with less than a hundred nodes. This criterion evaluates how well a solution will work when the size of the graph increases.

The stability criterion in the patrolling task measures the variation on the idleness of each node. Besides considering the *average idleness,* in some situations it is necessary to guarantee that there is no node with idleness much higher than the others.

The reactiveness criterion evaluates the approaches' capacity of patrolling in various worlds of different topologies without the need of learning time, recalibration of parameters and definition of specific strategies to perform the task in each graph. Finally, adaptability evaluates the ability of the approach to deal with graph modifications during simulation. Such a criterion is, for instance, applicable in computer games, where, in order to simulate real worlds, it can be necessary to do changes in the world, for example, removing an edge, simulating a construction of a wall or the destruction of a bridge. As well as removing a node in the case of a watching point being destroyed or taken by the enemy.

3 Previous Work

The first work on this task considered the problem of patrolling in non-weighted graphs (e.g. graphs where distance between adjacent nodes is one) [1]. The implemented solutions were simple, but covered several multi-agent approaches with varying parameters such as agent communication (allowed vs. forbidden), coordination scheme (central and explicit vs. emergent), agent perception (local vs. global), etc. The agents that presented better results in this work were *Conscientious Reactive* and *Cognitive Coordinated.*

Then, the real distance between nodes was considered, representing them as weights on the edges of the graph [2]. With this representation more sophisticated (non-adaptive) solutions that employed heuristically defined utility functions based on node idleness values and path lengths were explored. The agent that presented better result in this approach was *Heuristic Pathfinder Cognitive Coordinated (HPCC).*

Also, adaptive agents that learn to patrol using Reinforcement Learning (RL) techniques, mapping the patrolling problem into the standard RL framework, namely the

Markov Decision Process (MDP) formalism [4] were developed. This consisted in defining a state and action space for each agent, and in developing proper models of instantaneous rewards. The Q-Learning [4] algorithm was used to train individual agents. The *Gray-Box Learner Agent (GBLA)* is the best agent of this approach.

Another work, based on operation research, presented a single-cycle approach[3]. In this work, it is considered that an agent is capable of discovering a cycle in the graph to be patrolled, using solutions to the *Traveling Salesman Problem (TSP)*[3]. Then, the patrolling task consists in making the agents, equally distributed in space, follow this cycle repeatedly. We call *Single-Cycle (SC)* the agents following this TSP single-cycle strategy.

Negotiator agents were presented in [2]. In this work, each of the agents in a multi-agent system receives some nodes to patrol. During patrolling they can negotiate these nodes in order to split the graph into regions. Agent negotiations have been implemented as auctions. The negotiator agents which present better results in [2] were *Two-Shot-Bidder Agent* and *Mediated Trade Bidder Agent*.

A first comparison between all these works was presented in [2]. In that work, regarding average idleness criterion, the *Single-Cycle (SC)* was considered the most efficient agent. Despite the fact that the negotiator agents presented in [2] only achieved a good performance in two of the maps experimented (which will be shown in section 5), they do not suffer from problems like needing of tuning for each situation, nor do they require nor training periods, which can be prohibitive in huge graphs. Furthermore, they do not require centralized and explicit coordination schemes, which were critical to the good performance of the *HPCC*, for instance. In fact, Negotiator MAS are adaptable systems that can handle graph with priority regions and changes in topology during simulations. Although in [2] the negotiator agents did not presented such great results when one considered idleness standard deviation, there was a feeling that if the negotiators could do better partitions on the graph, this approach would be more stable than the others. In this light, we decided to improve the negotiator agents and run them in larger worlds in order to test if they could be scalable, stable, reactive and adaptable. Last, but by no means least, we also wanted to improve their average idleness results. In the next section we present details of how the negotiator approach has evolved.

4 Communicating and Negotiating

This work presents new types of negotiator agents and the bottom-up process of development of these types. First, the results when the utility function of these agents still did not consider the idleness of their nodes, only the distance between them will be shown. Then, we enhance this utility function considering also the average idleness of the nodes, allowing a more detailed view of the global situation when choosing a node to exchange at an auction.

4.1 Previous Negotiator Agents

In [2], the communication restrictions of the previous works were removed, allowing the agents to exchange messages with each other. Auctions are very popular for being

simple interaction scenarios, which makes them good as a first choice for helping agents to reach agreements [5, 6]. Independent agents can make use of auctions to distribute tasks and resources among them. Each negotiator agent receives randomically some graph nodes to patrol. The utility function of those agents was the negative of the distance it has to walk when visiting all the nodes they were responsible for patrolling. At each cycle of the simulation, each agent will search for the node that is decreasing its utility. Once the node is found, the agent tries to negotiate it. The agents can negotiate their nodes by making auctions in order to change their worst node for a better one.

4.2 New Agents

There are four major differences between the negotiators that will be presented here: (1) the auction protocol; (2) how they choose the node to be auctioned or to bid for; (3) their *path finding* algorithm; and (4) their behavior while individuals in a group.

The auctions can be one or two shots, private value and sealed-bid, which means that the auction can have one or two rounds, the value of the node on auction depends on the utility function of the agent and the bidder does not knows the bid of the others. Also, depending on the auction protocol, the agents can exchange, one node for another, two for one, or two for two.

The agent is responsible for patrolling a set of nodes received in the beginning of the simulation. Whilst in [2] the distribution of the nodes all over the graph was random, in this work neighbor nodes are randomically distributed. In order to guarantee the minimum time between two visits to the same node, the agent will try to minimize the distance between its nodes. So, each agent will try to get a set of nodes very near from its other nodes. When the agent detects its farthest node, it calls the others to an auction. The other agents will see if there is a node in their own set that can be exchanged by the one being offered, which means the exchange will make the bidder's nodes distance smaller. If there is such a node, this agent offers it as a bid. The auctioneer then chooses the best bid, which is the node that is nearer from its other nodes and makes the exchanging with the bidder.

In order to determine which node should be auctioned, we have first tested using a simple sort algorithm (insertion sort), where we calculate the distance the agent has to walk to visit its nodes, removing each node and recalculating the distance in order to find the one that increases most the path; then we tried the *barycenter* to determinate in this case the farthest node from *barycenter*. Also, we have tested the choice algorithm to be done with the Kruskal[7] algorithm to find the minimum cost spanning tree (MCST), since it is proven that MCST cost is at most two times the cost of the TSP [8]. The first version which uses insertion sort presented better results.

Some different ways of choosing the path of the agents were tested too, as walking in cycles by the agent nodes following the path created by the insertion sort, walking to the agent node with highest idleness, walking to the agent node with highest idleness considering distance, as the HPCC [2] and walking by the MCST. The algorithm presented at [2], walking to the agent node with highest idleness considering distance, presented the best result.

The agent's behavior can be self-interested or cooperative. In a cooperative negotiator MAS, agents can accept to trade one of its nodes by other that decreases its

utility function if the utility function of the group (which is the sum of the utilities of each agent) increases with this exchange.

We also developed MAS that made use of mediator agents, who worked as brokers that could help agents to find good opportunities of exchanging, but they have the disadvantage of having a bottleneck, which could cause the auction system failure, although the patrolling would continue to be done even if the mediator stops working.

Even the cooperative agents have a limitation which concerns exchanging: they will always bid their worst node, and sometimes exchanges are still needed but they do not happen because the agents keep offering the same node, since the utility function does not change. To facilitate node exchange, each agent randomically chooses a node after 20 cycles without exchanging. We have also tested building a priority queue of exchangeable nodes of each agent.

4.3 Agents Description

Following an incremental process, we have investigated the patrolling task with various market-based MAS approaches. The first type of agents developed was the *Bidder Agent*. These agents are self-interested, disregarding the group utility function.

Then a *Cooperative Bidder Agent (CBA)* was developed. CBA try to maximize the group utility. These agents consider the utility function of the bidder, as well as their own. Thus, a CBA will accept an exchange even if it decreases its utility function, but keeps the average utility highest.

These two types of agents are the basis for all other negotiator types created. They perform a simple one shot auction and exchange only one node. They choose nodes to auction or visit regarding the insertion sort algorithm. Since the *Bidder Agent* presented better results, other agents were developed based on it.

In [2], two patrolling negotiator agent types were shown:

1. Two-Shot-Bidder Agent (TSBA) – bids for nodes that increase its utility function. It promotes a two-shot auction, in an attempt to get better offers.
2. Mediated Trade Bidder Agent (MTBA) – in this architecture, there is a broker agent that informs the auctioneer which agents can be good bidders and which nodes they can trade.

In this work, we remove the restriction on the number of nodes of each agent can exchange at a time, which was static, and present the *Flexible Bidder Agent* (FBA) – like the *Bidder Agent,* these agents will change their worst node considering only their own utility function. They differ in the number of nodes exchanged. These agents can exchange up to two nodes with others, which means the number of nodes theses agents have to patrol can change during simulation. They can exchange two nodes by other two, two nodes by one, or one-by-one. Since the fundamental issue in deciding which nodes to exchange at a time is the agent's utility function, next section described how it would be better calculated.

4.4 Considering Idleness

In this work, we enhance the utility function of the negotiator agents by considering idleness and distance. Each agent uses the average idleness of its nodes to decide, so it

keeps the 10th last idleness observed in each node it has to patrol. As idleness and distance are different standards of measurement, it was necessary to use the idea of conversion of different scales, so the utility function now is the negative of the sum of the each agent node value, and node value is calculated as shown in equation (1).

$$nv = \alpha[(I - idleness)/(I - i)] + (1- \alpha)[(distance - d)/(D - d)]. \quad (1)$$

In the equation above, I is the highest average idleness between all agent nodes, i which is the low average idleness, D which is the longest edge the agent has to walk by and d the smallest. So, the agent will auction the node with the highest value.

5 Experimental Results

This section presents the methodology used in order to make possible comparisons between the best MAS presented in [2]. It details the set-up of the experiments, the proposed scenarios, and discusses the obtained results. The agents will be compared following the criteria proposed in section. 2 and average idleness proposed in [5].

5.1 Experiment Setup

All experiments were carried out using the same simulator and experiments used in [1, 2, 4]. The experiments were performed on six maps, which are shown in Fig. 1. Black blocks in the maps represent obstacles. White blocks are the nodes to be visited. Each MAS was simulated with populations of five and fifteen agents. For each pair (map, approach) we ran 10 experiments. At each experiment, all the agents start at the same node, but this starting node varies across the 10 experiments. Each simulation runs for 15000 cycles.

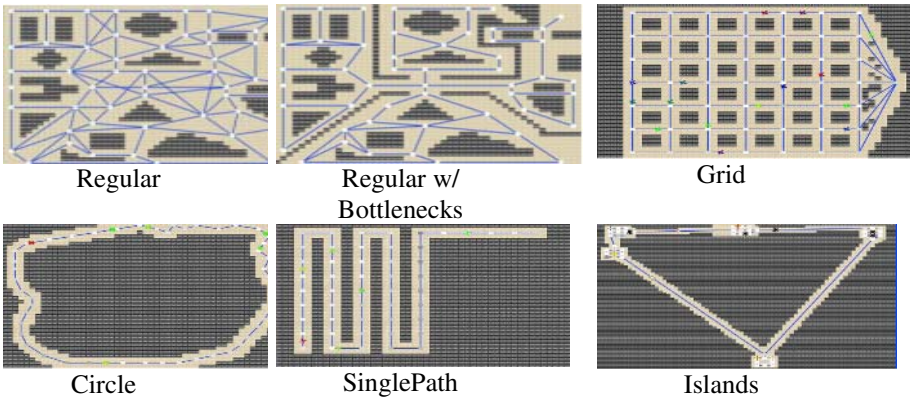


Fig. 1. Maps with different corresponding graph topologies on our simulator

5.2 Obtained Results

In the following graphics each colored bar represents a different MAS approach. First, the results obtained with the agents proposed in section 4.1 are shown. The agent

which shows the best results will be compared with the best agents of previous works. We noticed that the MAS where the agents are more cooperative presented worst results in most maps, as seen in Fig. 2 and Fig. 3. As explained in section 4, the agents compared in this section choose a random node after 20 cycles without exchanging and choose to visit their node with highest idleness considering distance.

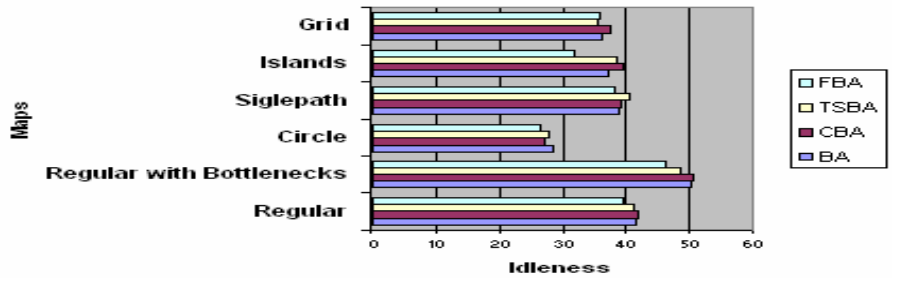


Fig. 2. Comparison of the average idleness performance of *Flexible Bidder Agent* (FBA), *Two-Shots Bidder Agent (*TSBA*), Cooperative Bidder Agent* (CBA) and *Bidder Agent* (BA) in simulations with 5 agents

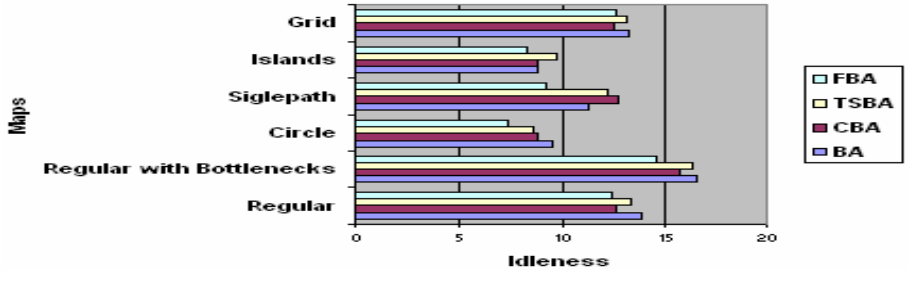


Fig. 3. Comparison of the average idleness performance of *Flexible Bidder Agent* (FBA), *Two-Shots Bidder Agent (*TSBA*), Cooperative Bidder Agent* (CBA) and *Bidder Agent* (BA) in simulations with 15 agents

Fig. 2 and Fig. 3 show the results for all maps and negotiator agent approach, for populations of 5 and 15 agents, respectively. The x-axis indicates the average graph idleness: the smallest the idleness, the better is the performance of the set of agents.

It can be seen in these graphics that *Flexible Bidder* (FBA) obtains the best performance in all cases, except in the grid map with 5 agents, where *Two-Shots* (TSBA) presents an equivalent result. So, we will compare FBA results with those of previous works. These agents still do not consider idleness in their utility function.

After finding the FBA as the best agent between negotiators, we then, improved its utility function, which now uses idleness, as shown in section 4.4 and compared it with previous agents of other approaches. The results were very good, as shown in Fig. 4 and Fig. 5.

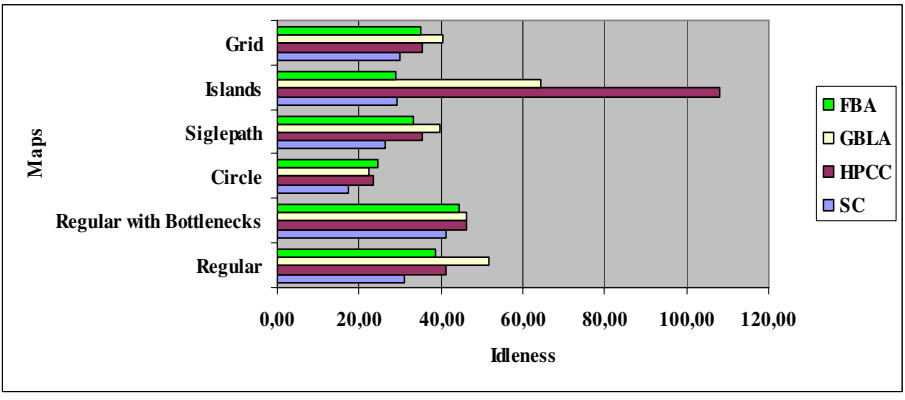


Fig. 4. Comparison of the average idleness performance of Flexible Bidder Agent (FBA), Gray-Box Learner Agent (GBLA), Heuristic PathFinder Cognitive Coordinated (HPCC) and Single-cycle Agent (SC) in simulations with 5 agents

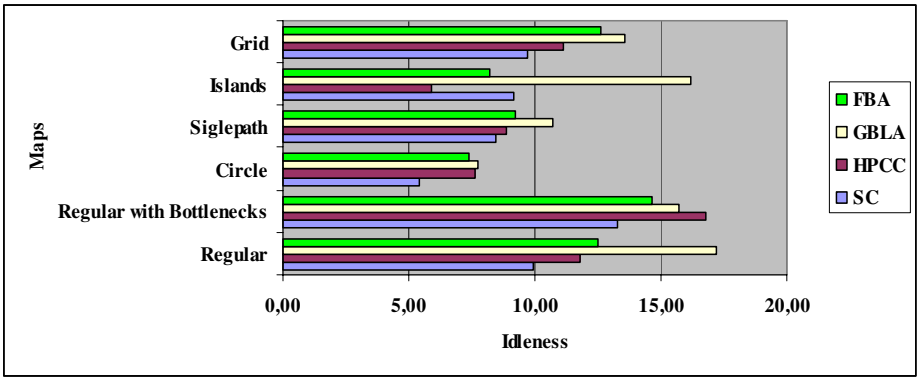


Fig. 5. Comparison of the average idleness performance of *Flexible Bidder Agent* (FBA), *Gray-Box Learner Agent (*GBLA*), Heuristic PathFinder Cognitive Coordinated (HPCC)* and *Single-cycle Agent* (SC) in simulations with 15 agents

As previously shown, it is clear from these graphics that Single Cycle (SC) obtains the best performance for all cases in *average idleness*, except in the islands map with 15 agents (where HPCC and FBA win). Regarding the FBA, for 5-agent simulations, FBA has the second best performance in all maps except in the circle. For 15 agents the FBA was the second best in two maps: circle, and regular with bottlenecks.

The Negotiator MAS do not depend on coordinators as the SC and the HPCC, which means they are more fault tolerant. The failure of the coordinator on the SC (before simulation starts) and HPCC (during simulation) MAS will make the system stop working. The *Flexible Bidder Agent* presented better results than *GBLA;* the other distributed approach, in all test scenarios and has the advantage of not needing the learning time. The Negotiators MAS are also more stable. With respect to graph

topology influence, while SC, GBLA and HPCC exhibit an equivalent performance variation (the standard deviation varies from 0.77 to 0.99) [2], the FBA presented standard deviation varying between 0,05 to 0,35.

Concerning the exchanges performance, it is shown in Fig. 6 that by the 1000th cycle there is no more exchanges, so by this cycle the agent node groups are stable.

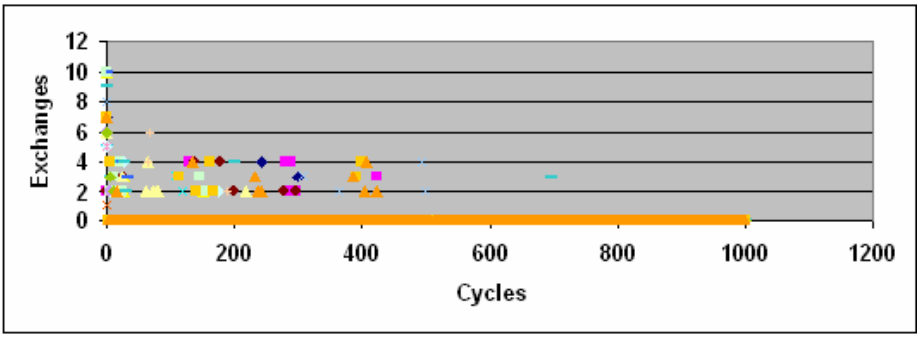


Fig. 6. Exchanges distribution through simulation

Considering the average idleness in the utility function was important, it presented results up to 20% better than without idleness for simulations with 5 agents. However, in maps with 15 agents, adding idleness to the utility function was at most 5% better. With regards to scalability, the negotiator agents can run in worlds of any size. The *FBA* was tested in a world with 200 and other with 500 nodes; the results are shown in Fig. 7. The HPCC could not run in a world of this size due restrictions in its algorithm; the *GBLA* could not be simulated either, since it requires too much time for learning. We believe that the same problem of time shall be the case of the SC, however it was not tested.

Regarding reactiveness, the FBA has shown good results, as all negotiator agents need no learning time, nor preparation of a new cycle to be capable of running in each map. In this respect, the HPCC also does not need preparation for each map. Adaptability was not tested, but it is clear that the SC would have to stop simulation and find other cycle for continuing simulation; the GBLA would also require extra time for stabilization.

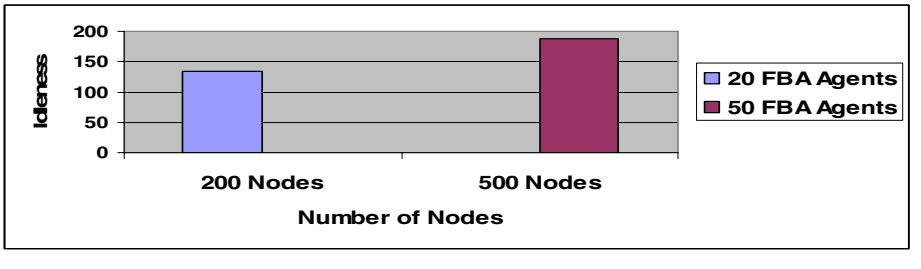


Fig. 7. Simulation results in large worlds

6 Conclusions and Further Work

Multi-agent patrolling is a task that can be useful in different domains (e.g. games), and, as such, has been approached with various different techniques. However, none of them has proved to be the best solution for all domains and scenarios. Thus, in this work, we suggest a set of criteria to better analyze the approaches to patrolling, helping us determine when to choose each technique. Based on those criteria, we have chosen to deepen our previous studies in negotiation-based multi-agent patrolling since the approaches that presented better results in previous works have problems with training costs or centralized and explicit coordination The negotiation-based approaches presented in this work are distributed, stable, reactive, adaptable and scalable. The experimental work we have carried out with the proposed negotiation based approaches have yielded good results when compared to those available so far on patrolling. We explored the agent communication ability to implement auction mechanisms, which provide good agent coordination when partitioning the graph to be patrolled.

There are still several open issues concerning the research on negotiation-based multi-agent patrolling. We are already working on augmenting the complexity of the agents and MAS, including partitioning of the graph by means of graph theory algorithms and using heterogeneous systems, combining various approaches on same simulation. We are also interested on performing new experiments to better evaluate the approaches with respect to adaptability.

References

1. Machado, A., Ramalho, G., Zucker, J., Drogoul A.: Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures. Multiagnet Based Simulation. Bolona, Itália. (2002)
2. Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V., Chevaleyre, Y: Recent Advances on Multiagent Patrolling. The 17th Brazilian Symposium on Artificial Intelligence. Springer Verlag. (2004). 474-483
3. Chevaleyre, Y., Sempé, F.: A theoretical analysis of multi-agent patrolling strategies. Poster at the International Conference on Autonomous Agents and Multi-Agents Systems (AAMAS) (2004)
4. Santana, H., Ramalho, G., Corruble, V., Ratitch, B.: Multi-agent Patrolling with Reinforcement Learning. International Conference on Autonomous Agents and Multi-agents Systems (2004)
5. Faratin, P., Sierra, C., Jennings, N.: Negotiation Decisions Functions for Autonomous Agents. (1998)
6. Faratin, P., Sierra, C., Jennings, N.: Using Similarity Criteria to Make Issue Trade-Offs in Automated Negotiations. Artificial Intelligence Journal (2002).
7. Manber, U.: Introduction to Algoritms - A Creative Approach, Addison-Wesley, Tucson, (1989). 208 - 211
8. Helvig, C., Robins, G., Zelikovsky, A.: Moving Target TSP and Related Problems. Proceedings of the 6th Annual European Symposium on Algorithms (1998). 453-464

Running Agents in Mobile Devices

Javier Bajo[1], Juan M. Corchado[2], and Luis Fernando Castillo[3]

[1] Universidad Pontificia de Salamanca
Compañía 5, 37002, Salamanca, Spain
jbajope@upsa.es
[2] Departamento Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
corchado@usal.es
[3] Dep. Ingeniería Industrial, Universidad Nacional de Colombia
Sede Manizales – Colombia, Campus la Nubia
lfcastilloos@unal.edu.co

Abstract. This paper presents a deliberative architecture based on the concept of CBP-BDI agent. A CBP-BDI agent is a BDI agent that incorporates a CBP reasoning engine. The work here presented focuses in the development of a multiagent system that has been constructed for the management of some aspects of a shopping mall, specially the interaction with clients, so the aim is to get the portability of a CBP-BDI agent to mobile devices. The system has been tested and this paper presents the results obtained.

1 Introduction

Multiagent systems have become increasingly relevant for developing applications in dynamic, flexible environments, such as the internet, personalized user interfaces, oceanography, control systems or robotic. Agents can be characterized through their capacities such as autonomy, reactivity, pro-activity, social abilities, reasoning, learning and mobility. These capacities can be modelled in various ways, using different methodologies [18]. One of the possibilities is to use Case Based Reasoning (CBR). This paper presents a distributed architecture whose principal characteristic is the use of CBP-BDI agents. These deliberative agents incorporate a reasoning Case Based Planning (CBP) engine, a variant of the CBR system which makes it possible for the agents to learn from initial knowledge, interact autonomously with the environment and users, and allows it to adapt itself to environmental changes.

The aim of this work is to obtain an architecture that allows the development of agents which incorporate CBP reasoning mechanisms, for dynamic environments. To achieve this aim we have concentrated in a specific problem, the management of some aspects of a shopping mall, and we use an architecture that makes it possible to construct agents capable of adapting its knowledge to environmental changes. There are many different architectures for constructing deliberative agents and many of them are based on the BDI model. In the BDI model, the internal structure of an agent and its capacity to choose, is based on mental aptitudes. This has the advantage of using a natural model and a high level of abstraction. The BDI model uses the agent's beliefs as informational aptitudes, its desires as motivational aptitudes and its intentions as

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 58 – 67, 2006.

deliberative aptitudes. The method proposed in [6] facilitates the incorporation of CBR systems as a deliberative mechanism within BDI agents, allowing them to learn and adapt themselves, lending them a greater level of autonomy than pure BDI architecture [13].

The system incorporates "lightweight" agents that can live in mobile devices, such as phones, PDAs, etc. [7]. These agents make it possible for a client to interact with the MAS in a very simple way, downloading and installing a personal agent in his mobile phone or PDA. Case-Based Mark-up Language (CBML) [11], a XML based language for representing CBR components, is used in order to allow us to work within mobile devices. So the formal definition of the structure of our cases and similarity measures are completely independent of the application code and CBR components can be exchanged between heterogeneous CBR systems. The agents are adapted to work in mobile devices, so they support wireless communication (Wi-Fi, Bluetooth) which facilitates the portability to a wide range of mobile devices [7].

One of the major problems in the development of an architecture based on multiagent systems is that there are currently no clear standards or well developed methodologies for defining the steps of analysis and design that need to be taken. There are at present a number of methodologies: Gaia [19], AUML [3], INGENIAS [14], TROPOS [17], MESSAGE [12]. For this study, we have decided to opt for Gaia for our MAS. Gaia is a simple methodology that allows us to carry out a preliminary analysis and design with which to confront the problem at a general level. The great advantage is that we can carry out a rapid, broad study. we are able to obtain both a generalized vision of the problem in terms of organization, which helps enormously in the development of such a research project.

In the next section, we will explain the shopping mall problem that has led to most of this research. In the third section we will describe the wireless multiagent system developed. In the fourth section, the case based planning system will be presented. Finally, some preliminary results and the conclusions will be presented.

2 A Case Study

The Mall has become one of the most prevalent alternative to traditional shopping [2]. A shopping mall is a cluster of independent shops, planned and developed by one or several entities, with a common objective. The size, commercial mixture, common services and complementary activities developed are all in keeping with their surroundings [2]. Every shopping mall has a permanent image and a certain common management. A shopping mall needs to be managed and, the management includes solving incidents or problems in a dynamic environment. As such, a shopping mall can be seen as a large dynamic problem, in which the management required depends on the variability of the products, clients, opinions, etc. Within this framework, the multiagent system technology developed in this project will make it possible to provide better services to the shopping mall clients. Our aim is to develop an open system, capable of incorporating as many agents as necessary, agents that can provide useful services to the clients not only in this shopping centre, but also in any other environment such as the labor market, educational system, medical care, etc. The system provides mechanisms for easy data consulting. A user will be able to gain

access to shopping and sales and leasing time information (entertainment, events, attractions, etc.) by using their mobile phone or PDA. Mechanisms for route planning when a user wants to spend time in the mall will also be available. Moreover, it provides a tool for advertising offers (a commercial manager will be able to make his offers available to the shopping mall clients), or provides a tool to the shopping mall management team in order to contact commercial managers or shopping mall clients, providing an interaction between users interested in the same topics.

3 SM_MAS: Shopping Mall Multiagent System

The option chosen to define an appropriate analysis and design methodology for the problem to be resolved is Gaia [19]. Gaia is a methodology for analysis and design very general and therefore applicable to a very wide range of multiagent systems. It also allows the user to have a wide knowledge of the multiagent systems both at an organizational (social) level and at a detailed level for each agent.

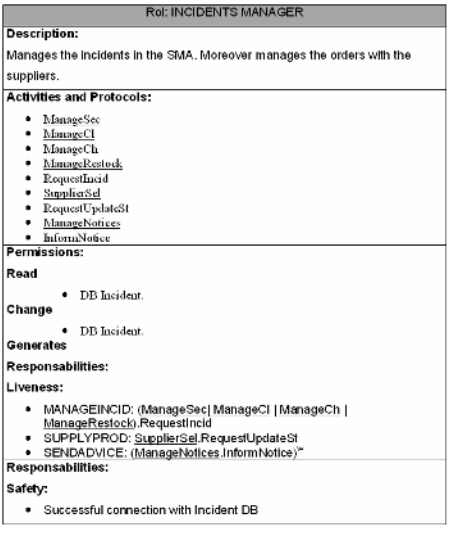


Fig. 1. Gaia roles model: Incidents Manager role

Through the Gaia analysis, two models are obtained: the role model and the interaction model. Studying the requirements of the problem we have come to the conclusion that we need nine roles: The Communicator role manages all the communications of a client. The Finder role looks for near devices. The Profile Manager role obtains a client profile. The Store operator is in charge of manage the store: data base operations on stored products. Moreover monitors the products shortage, in order to prevent desupply. The Promotions manager role controls the shells in each shop, as well as the promotions that every shop offers to its clients. The Clients Manager role deals with the client profiles management and controls the connected clients at a given moment. The Analist role carries out periodic evaluations on shells, promotions and

surveys data trying to provide a good quality service. The Incidents Manager role manages incidents, such as sending advices, or solving a wide range of problems. The Planner role is the most important role in our system. The Planner creates a route printing the most suitable shops, promotions or events to the client profile and available resources at one particular moment. As can be seen in Figure 1, the Incidents Manager role is composed of responsibilities, permissions, activities and protocols [19]. It is authorized to read and change the Incidents DB, and it is responsible for the incidents management, product orders and sending advices. Besides it must maintain a successful connection with the Incidents DB.

As far as interaction model is concerned, the dependences and relations between roles are described. Each interaction in which two roles are involved requires protocols (described in the roles model). In the SMA presented in this work the next protocols have been considered: RequestPromotionsData, SolveConsult, StoreProducts, AlertShortage, OrderSupplier, InformProductsState, InformPromotionsState, SolveIncident, SolveSuggestion, Notify.

Fig. 2. Gaia agent model for the shopping mall problem

Once the analysis has been finalised, the Gaia design is carried out. The level of abstraction is reduced so that traditional techniques can be applied. In the design process three models are considered: agents model, services model and acquaintance model [19]. As we can see in Figure 2, the agents model shows the types of agents that are going to appear in the system, as well as the number of instances for each agent type that can be executed within the execution time. For example agent Store plays the Promotions Manager and Store Operator roles. In order to obtain a more detailed design, AUML has been applied after the Gaia design.

4 Deliberative Behaviour

The purpose of case-based reasoning (CBR) is to solve new problems by adapting solutions that have been used to solve similar problems in the past. The CBP is a variation of the CBR which is based on the plans generation from cases. The deliberative agents, proposed in the framework of this investigation, use this concept to gain autonomy and improve their problem-solving capabilities. As described in [13], in a CBP-BDI agent, each state is considered as a belief; the objective to be reached may also be a belief. The intentions are plans of actions that the agent has to carry out in order to achieve its objectives [6], so an intention is an ordered set of actions; each change from state to state is made after carrying out an action (the agent remembers the action carried out in the past, when it was in a specified state, and the subsequent

result). A desire will be any of the final states reached in the past (if the agent has to deal with a situation, which is similar to a past one, it will try to achieve a similar result to the previously obtained one). Next, the CBP planner used for the Client Agent to find a route in the shopping mall is presented: Let $E = \{e_0,...,e_n\}$ the set of the possible interesting places to visit and buy.

$$a_j : \underset{e_i}{E} \rightarrow \underset{a_j(e_i)=e_j}{E} \tag{1}$$

An Agent plan is the name given to a sequence of actions (1) that, from a current state e_0, defines the path of states through which the agent passes in order to offer to the client the better path according to each client's characteristics. Below, in (2), the dynamic relationship between the behaviour of the agent and the changes in the environment is modelled. The behaviour of agent A can be represented by its action function $a_A(t) \ \ \forall t$, defined as a correspondence between one moment in time t and the action selected by the agent,

$$Agent\ A = \{a_A(t)\}_{t \in T \subseteq N} \tag{2}$$

From the definition of the action function $a_A(t)$ a new relationship that collects the idea of an agent's action plan (3) can be defined,

$$p_A(t_n) = \int_0^{t_n} a_A(t)dt \tag{3}$$

The variation of the agent plan $p_A(t)$ will be provoked essentially by: the changes that occur in the environment and that force the initial plan to be modified, and the knowledge from the success and failure of the plans that were used in the past, and which are favoured or punished via learning. O indicates the objectives of the agent and O' are the results achieved by the plan. R represents the total resources and R' are the resources consumed by the agent. The efficiency of the plan (4) is the relationship between the objectives attained and the resources consumed

$$E_{ff} = \frac{\#(O' \cap O)}{\#R'} \tag{4}$$

Where # means cardinal of a set. The objective is to introduce an architecture for a planning agent that behaves – and selects its actions – by considering the possibility that the changes in the environment block the plans in progress. This agent is called MRPI (most re-plan-able Intention agent) because it continually searches for the plan that can most easily be re-planned in the event of interruption. Given an initial point e_0, the term planning problem is used to describe the search for a way of reaching a final point $e_i \equiv e^* \in E$ that meets a series of requirements. Given a problem E and a plan $p(t)$ the functions Ob and Rc accumulated are constructed from the objectives and costs of the plan (5). For all time points t_i two variables are associated:

$$Ob(t_i) = \int_a^{t_i} O(t)dt \ \ Rc(t_i) = \int_a^{t_i} R(t)dt \tag{5}$$

This allows us to construct a space representing the environment for planning problems as a vectorial hyper dimensional space where each axis represents the accumulative

variable associated with each objective and resource. In the planning space, defined in this way, conform to the following properties:

1. Property 1: The representations of the plans within the planning space are always monotonously growing functions. Given that *Ob(t)* and *Rc(t)* are functions defined as positive, function *p(t)* expressed at these coordinates is constant or growing.
2. Property 2: In the planning space, the straight lines represent plans of constant efficiency. If the representations of the plans are straight lines, the slope of the function is constant, and coincides with the definition of the efficiency of the plan. $\frac{d}{dt} p(t) = cte \Leftrightarrow \lim_{\Delta \to 0} \frac{\Delta O(t)}{\Delta R(t)} = cte$

In an n-dimensional space, the extension of the straight concept line is called a geodesic curve. In this sense, the notion of geodesic plans can be introduced, defined as those that maintain efficiency at a constant throughout their development. This way, only the plans of constant efficiency (geodesic plans) are considered, due to the fact that they are the ones of minimum risk. In an environment that changes unpredictably, to consider any plan that is different from the geodesic plan means to accept a certain risk. The agent must search for the plan that determines a solution with a series of restrictions $F(O;R)=0$. In the plans base the plans sought are those that are initially compatible with the problem faced by the agent, with the requirements imposed on the solution according to the desires, and in the current state [1]. If all the possible plans $\{p_1,...,p_n\}$ are represented within the planning space, a subset of states that the agent has already attained in the past will be obtained in order to resolve similar problems. With the mesh of points obtained (generally irregular) within the planning space and using interpolation techniques, we can obtain the working hyperplan *h(x)* (that encapsulates the information on the set of restrictions from restored experiences, by definition leading to a hyperplan since it verifies $h(x_j)=p_j\ j=1,...,n$ and the planning space is the dimension n). From this, geodesic plans can be calculated and and the variation calculation is applied. Suppose, for simplicity's sake, a planning space of dimension 3 with coordinates $\{O,R_1,R_2\}$. Between point e_0 and objective points $f_s f=\{e_1,..., e_m\}$ and over the interpolation surface *h(x)*, the Euler Theorem [13, 15] guarantees that the expression of the geodesic plans will be obtained by resolving the system of equations in (6), where R_i is the function accumulated *R*, *O* is the function of accumulated *O* and *L* is the distance function on the hyperplan *h(x)*, $L=\int_h dl$..

In order to obtain all the geodesic plans that, on the surface *h(x)* and beginning at e_0, allows us to reach any of the points $e^* \in f_s f$, a condition of the surrounding must be imposed: the initial point will be $e_0=(O_0,R_0)$. Once an efficient plan is developed, the plan around it (along its trajectory) are used to create a denser distribution of geodesic plans. The tool that allows us to determine this is called the minimum Jacobi field associated with the solution set [17]. $g_0:[0,1]\rightarrow S$ be a geodesic over a surface *S*. Let $h:[0,1]\ x[-\varepsilon,\varepsilon]\rightarrow S$ be a variation of g_0 so that for each $t \in (-\varepsilon,\varepsilon)$, the set $\{h_t(s)\}_{t\in(-\varepsilon,\varepsilon)}$: $h_t(s)$ for all $t\in(-\varepsilon,\varepsilon)$ are geodesic in *S* and they begin at $g_0(0)$, in other words, they conform to $h_t(0)=g_0(0)$ for all $t\in(-\varepsilon,\varepsilon)$. In these conditions, taking the variations to a differential limit (7).

$$\begin{cases} \dfrac{\partial L}{\partial R_1} - \dfrac{d}{dO}\dfrac{\partial L}{\partial R_1'} = 0 \\ \dfrac{\partial L}{\partial R_2} - \dfrac{d}{dO}\dfrac{\partial L}{\partial R_2'} = 0 \end{cases} . \tag{6}$$

$$\lim_{t\to 0}\{h_t(s) = g_0(s+t)\} = \lim_{t\to 0}\{h(s,t)\} = \frac{\partial g_0}{\partial t}\Big|_{(s,0)} = \frac{dg_0}{ds} \equiv J_{g_0}(s) \tag{7}$$

The term $J_{g0}(s)$ is given to the Jacobi Field of the geodesic g_0 for the set $\{g_n(x)\}_{n\in N}$, and in the same way that the definition has been constructed, it is possible to give a measurement for the distribution of the other geodesics of $\{g_n(x)\}_{n\in N}$ around g_0 throughout the trajectory. Given a set of geodesics, some of them are always g* that, in their environment, have a greater distribution than other geodesics in a neighbouring environment. This is equivalent to saying that it presents a variation in the distribution of geodesics lower than the others and therefore the Jacobi Field associated with $\{g_n(x)\}_{n\in N}$ reaches its lowest value at J_{g^*}. Let's return to the MRPI agent problem that, following the recuperation and variation calculation phase, contains a set of geodesic plans $\{p_1,...,p_n\}$. If the p^* is selected with a minimum Jacobi Field value, it can be guaranteed that in the event of interruption it will have around it a greater number of geodesic plans in order to continue. This suggests that given a problem with certain restrictions $F(O;R)=0$, the geodesic plan p^* with minimum associated Jacobi field associated with the set $\{g_n(x)\}_{n\in N}$ is called the most re-plan-able solution. The behaviour model G for the MRPI agent is (8).

$$G(e_0, p_1, \cdots, p_n) = p^* \Leftrightarrow \exists n \in N / J_{g_n} \equiv J_{g^*} = \underset{n\in N}{Min} J_{g_n} \tag{8}$$

If the plan p^* is not interrupted, the agent will reach a desired state $e_j \equiv e^* \in f_s f_j \in \{1,...,m\}$. In the learning phase, a weighting $w_f(p)$ is stored. With the updating of weighting $w_f(p^*)$, the planning cycle of the CBP motor is completed. In Figure 3, it is possible to see what happens if p^* is interrupted. Let's suppose that the agent has initiated a plan p^* but at a moment $t>t_0$, the plan is interrupted due to a change in the environment. The geodesic planning meets the conditions of the Bellman Principle of Optimality [5], in other words, each one of the plan's parts is partially geodesic between the selected points. This guarantees that if g_0 is geodesic for interrupted e_0 in t_1, because e_0 changes to e_1, and g_1 is geodesic to e_1 that is begun in the state where g_0 has been interrupted, it follows that: $g = g_0 + g_1$ is geodesic to $e = e_0 (t_1 - t_0) + e_1 (t_2 - t_1)$.

The dynamic process follows the CBP cycle recurrently: each time a plan finds itself interrupted, it generates from the state reached so far, the surroundings of the plans from the case base and adjusts them to the new problem. With this it calculates the geodesic plans and selects the one which meets the minimum conditions of the associated Jacobi field. In this way the dynamic planning model of the agent G(t) is characterised as it is shown in the Figure 4. A minimum global Jacobi field J(t) also meets Bellman's conditions of optimality [5], in other words, a minimum global Jacobi field, must select minimum Jacobi fields "in pieces" (9).

$$J_{\min}(t) = \{J_{\min}(t_1 - t_0), J_{\min}(t_2 - t_1), \cdots, J_{\min}(t_n - t_{n-1})\} \tag{9}$$

If on the one hand, successive Jacobi fields generate one Jacobi field, and on the other hand, minimum Jacobi fields generate a minimum Jacobi field, the MRPI agent that follows a strategy of replanning *G(t)* as indicated to survive a dynamic environment, generates a global plan $p^*(t)$ that, faced with all possible global plans $\{ p_n(t)\}_{n \in N}$, presents a minimum value in its Jacobi field $J_{g^*}(t) \equiv J_{p^*}(t)$. An agent has been formally defined that in a dynamic environment seeks plans that lend it greater capacity for replanning.

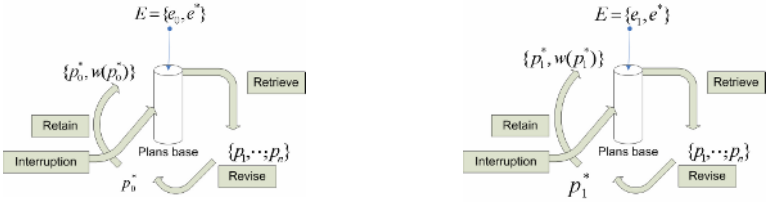


Fig. 3. Model for behaviour G(t)

5 Results and Conclusions

The previously described system was tested at the Tormes Shopping Mall in the city of Salamanca during 2005. During this period of time, the multiagent system has been tuned and updated, and the first autonomous prototype started to work in October 2005. Although the system is not fully operational and the aim of the project is to construct a research prototype and not a commercial tool, the initial results have been very successful from the technical and scientific point of view. The construction of the distributed system has been relatively easy using previously developed CBR-BDI libraries [4, 8, 9, 10]. Gaia [19] provides an adequate framework for the analysis and design of distributed agent based systems. The formalism defined in [15] facilitates the straight mapping between agent definition and CBR construction.

The fundamental concept when we work with a CBR system is the concept of case, and it is necessary to establish a case definition. A case structure in our problem, managed by the Client agent, is defined using CBML [11] as can be seen in Figure 4. The structure is defined through the feature labels. At execution time the client profile cases are instantiated as shown in Figure 4. The items, attributes and their values and weights are labelled. In our problem three main attributes have been considered: personal data, retail/leisure time data and interests data. The retail/leisure attribute is composed of business type, business identification, product type, product identification, price, units and date attributes. The interests data attribute is composed of retail time and frequency, monthly expense both business and product, extracted from retail data, and the explicit attributes obtained from questionnaires. Each attribute has a value, noun or adjective, and an assigned weight. Since the number and type of business is high, the businesses were classified into leisure time (cinema and recreational), catering (restaurant, fast food and pizza) and public retail (clothes, shoes, computing, supermarket and optical). The products have been also classified.

The system implementation has involved an increase in benefits due to the generation of automatic promotions. The e-commerce techniques [2] have facilitated custom-designed services for the clients. A user can easily find the products he is

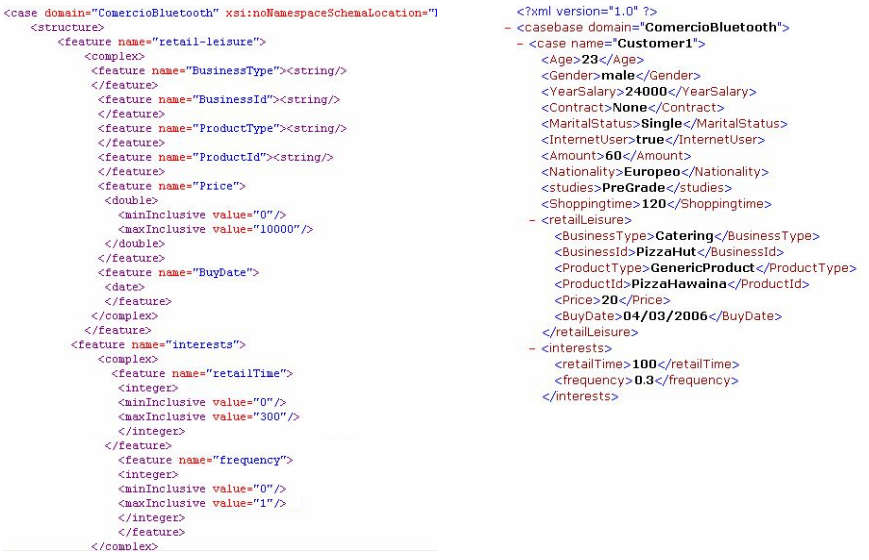


Fig. 4. CBML Cases structure and case instante

interested on, spend his leisure time in a more efficient way and make contact with other clients with whom he can share hobbies or opinions. So the degree of client satisfaction has been improved as observed in the surveys. The percentage of the sale of promotional products has grown over the total. The fundamental reason is that clients have instantaneous information about the products the agent thinks they are interested on, and the information is very accurate and customized.

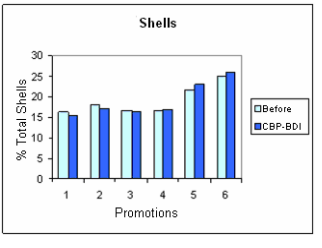


Fig. 5. Shell promotional products and shell total products

The increase in promotional products sold can be seen in Figure 5 that shows the increased percentage in promotional products sold in comparison with the promotional products sold using traditional commercial techniques carried out the year before. Selected products were those of constant demand to avoid the influence of external factors. We can observe that at the beginning, the results obtained with the multiagent system were worse than traditional techniques. However, as the system obtained more information about client profiles, products and habits, so the knowledge it obtained became more suitable and it was able to create optimal plans. Moreover the clients also needed some time to get used to the new system.

References

1. Aamodt A. and Plaza E.: Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7., pp 39-59 (1994)
2. Adams, F.G.: The E-Business Revolution & the New Economy: E-conomics after the Dot-Com Crash. South-Western Educational Pub. (2003)
3. AUML. www.auml.org (2006)
4. Bajo J. and Corchado J.M.: Evaluation and monitoring of the air-sea interaction using a CBR-Agents approach Proccedings of the 6th Internacional Conference on Case-based Reasoning, ICCBR'05 pp. 50-62. Springer Verlag. (2005)
5. Bellman, R.E.: Dynamic Programming. Princeton University Press, New Jersey (1957)
6. Bratman, M.E.: Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A. (1987)
7. Corchado J.M., Corchado E. y Pellicer M.A.: Design of cooperative agents for mobile devices. Proceedings of the International Conference on Cooperative Design, Visualization and Engineering - CDVE2004. Luo Y. (Ed.) LNCS 3190, Springer Verlag. ISSN 0302-9743. ISBN 3-540-23149-8 pp. 205-212. (2004)
8. Corchado J. M. and Laza R.: Constructing Deliberative Agents with Case-based Reasoning Technology, International Journal of Intelligent Systems. Vol 18, No. 12, December. pp.: 1227-1241 (2003)
9. Corchado J. M. and Lees B.: A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence. Vol 15, no. 2, pp.105-127 (2001)
10. Corchado J. M., Pavón J., Corchado E. and Castillo L. F. : Development of CBR-BDI Agents: A Tourist Guide Application. 7th European Conference on Case-based Reasoning. LNCS 3155, Springer Verlag. pp. 547-559 (2005)
11. Coyle, L., Cunningham, P. & Hayes, C.: Representing Cases for CBR in XML. Proceedings of 7th UKCBR Workshop, Peterhouse, Cambridge, UK (2002)
12. EURESCOM MESSAGE: Methodology for engineering systems of software agents. Technical report P907-TI1, EURESCOM (2001)
13. Glez-Bedia M. and Corchado J. M.: A planning strategy based on variational calculus for deliberative agents. Computing and Information Systems Journal. Vol 10, No 1, 2002. ISBN: 1352-9404, pp. 2-14 (2002)
14. Gomez, J. and Fuentes, R.: Agent oriented software engineering with INGENIAS. In Proceedings of 4th Iberoamerican Workshop on Multi-Agent Systems (Iberagents'02), Malaga, Spain. (2002)
15. Jost, J. and Li-Jost, X.: Calculus of variations. Cambridge University Press, U.K (1998)
16. Lee, J.M.: Riemannian Manifolds. An introduction to Curvature. Springer-Verlag, New Tork, Inc (1997)
17. Mylopoulos, J., Kolp, M., and Castro, J.: UML for agent-oriented software development: The TROPOS proposal. In Proc. of the 4th Int. Conf. on th Unified Modeling Language UML'01, Toronto, Canada. (2001)
18. Wooldridge, M. and Jennings, N. R.: Agent Theories, Architectures, and Languages: a Survey. In: Wooldridge and Jennings, editors, Intelligent Agents, Springer-Verlag, pp. 1-22 (1995)
19. Wooldridge, M. and Jennings, N. R. and Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems, 3 (3). pp. 285-312. (2000)

A Multi Agent Based Simulator for Brazilian Wholesale Electricity Energy Market

Wagner da Silva Lima and Eduardo Noronha de Andrade Freitas

Energy Research and Studies Workgroup, Electrical and Computer Engineering School, UFG, Bloco A, 2° andar, Praça Universitária s/n, CEP 74.605-220 Goiânia, Goiás, Brazil
{wagner, noronha}@eee.ufg.br
http://www.eee.ufg.br

Abstract. During the last two decades power systems industry has undergone many changes world-wide, seeking to create competitive wholesale electricity markets. In these markets, power suppliers and consumers are free to negotiate the terms of their contracts, e.g., energy prices, power quantities and contract durations in several auctions structures. Multi-agent systems are a widely used computer paradigm in the simulation of these new power market business models. A usual approach is to develop a specific application within a standard agent management framework, such as JADE. Several approaches are commonly used for market simulations: those that evaluate the behavior of supplier and demand agents under several auction models, and others that focus on each market's particular operations and problems, i.e., they simulate real-time market operating conditions. This work presents a multi-agent-based simulator for Brazilian electrical power market, incorporating operational and commercial models.

1 Introduction

Privatization of the Latin American power industry began in 1982 in Chile and reached Brazil in 1995. The main objective was the implementation of competitive wholesale energy markets, reflecting a world-wide industry trend.

The main motivation behind the transition to a competitive power market in Brazil was to increase private investment share in generation and transmission assets, thereby assuring system growth. There were many similar wholesale energy market transitions world-wide. The main characteristics of this new commercial structure are:

- *Competition between generation and consumer agents*, who freely negotiate electricity energy quantities through bilateral contracts and spot markets;
- *Electricity transmission grid* (transmission and distribution companies) are considered a natural monopoly and permits *open access* by payment of connection and use charges (regulated tariffs).

The operational (system dispatch and security), commercial (bilateral contracts) and regulatory (market rules) components, as well as the methodologies and structures of the power industry form a complex model for simulation. In order to analyze the agents´ roles, a new paradigm is required. Many analyses of this complex theme may be found in the literatute. These approaches may be classified into two

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 68 – 77, 2006.

categories: (a) models that simulate auction structures and rules, focusing on agent learning strategies as a way of maximizing its benefits; and (b) models that simulate real-time system conditions, electric operation security assessment, equipment outages, energy management systems, expansion planning analysis and generation scheduling.

Multi-agent systems have been *strongly* used to simulate competitive power markets. LIU (2001) [1] built a general-purpose power simulator based on MAS. Prior to that simulator, only a few power simulators were developed. Recently many models appeared in the literature [2]. However, none incorporate the tight pool market structure required to simulate Brazilian wholesale energy market conditions. Two papers present generation bidding strategies learning process without detailed simulator model.

Ferreira (2001) [3] present a MAS application to a tight pool wholesale market in a small hydrothermal system. The Hydropower system portion is dispatched by an optimization model with an objective function that minimizes the future expected generation cost employing a stochastic dynamic programming method. Thermal power plants agents participate in an auction model offering energy volume and prices for the residual energy in order to reach the total demand. The thermoelectric generation agent decision process was modeled with non-cooperative games and Nash-Cournot equilibrium criteria. A custom management agent framework in JAVA language was constructed.

Carvalho (2004) [4] implemented a purely thermoelectric wholesale energy market to evaluate reinforced learning performance for bidding strategies. The JADE (Java Application Development Environment) [5] MAS framework environment was used to manage the agents.

This work presents a new tight pool wholesale energy market simulator called DYNAMIS, concerning Brazilian specific operational and commercial aspects using MAS technology, JADE framework, JAVA language and an object oriented programming paradigm. With DYNAMIS a more realistic simulation of the Brazilian Wholesale Energy Markets (BWEM) was achieved. Braziliam tight pool market has peculiar characteristics with strong transmission interconnected system and many hydrographic basins too interconnected with great inflows variability with predominantly hydroelectric generation. Those features were not modeled in a MAS approach previously in literature.

The rest of this paper is organized as follows. Section 2 presents tight pool wholesale energy markets' main aspects. Section 3 presents a multi-agent system overview. Section 4 presents the proposed multi-agent simulator. Section 5 presents a case study. Section 6 contains conclusions and describes interesting issues for future extensions.

2 Tight Pool Market Overview

Wholesale energy markets have a structure organized by a spot market and a long term market (forward or bilateral contracts market). Long term contracts are described by three main attributes: volume, price and duration. Sellers and buyers of energy negotiate freely the terms and forms of the contracts' attributes.

A spot market represents the short term 24-hour look-ahead market conditions in which prices and generation dispatch are defined. In general there are two types of

spot markets. In the *loose pool market* the commercial ("ex-ante") generation dispatch is carried out through auctions where generators and demand agents bid for price and/or quantities. A Power Exchange agent (PX) is necessary to process bidding and defines spot price.

Another type of spot market is defined where the commercial dispatch is centralized. Aiming for maximization of the energy production by the generating system as a whole, the operation is coordinated by an Independent System Operator (ISO), in a "tight pool" mode, where the generation plants dispatch does not depend on the power plant owners, but instead depends exclusively on the ISO decisions. This type of market is called a *tight pool market*. Spot price reflects the generation short range marginal price.

The agents celebrate bilateral contracts with each other. PX agents deal only with non-contracted energy. This energy volume, not bound to contracts is bought or sold by spot price. This wholesale market structure is called market of differences. Thus, the short term and long term markets work in parallel and are represented in PX.

The Brazilian electric generation system has the peculiarity of being basically hydroelectric with multiple reservoirs of large storage capacity. Therefore, its production capacity is essentially stochastic in nature. The many factors which affect the operational behavior of hydro and thermal plants are fundamental to finding the best ways to simulate the plant´s operation.

The Brazilian power generation system operates with a thermal complementation mode. This type of operation has the objective of dispatching the thermal plant with technical support, making use of powerful optimization methods to reach optimal operation of the hydrothermal generating system. The result of this process is the future expected cost function and the immediate operation cost function. These functions permit the simulation of the generation dispatch by an ISO. This means that the plant will only be dispatched when the electric power system requires thermoelectric power to complement the hydro energy deficits derived from the lack of water in the reservoirs of the hydroelectric system.

The natural inflow variability is characterized by a stochastic process, causing spot prices volatility at the BWEM, depending on hydrology. Periods with large inflows (wet period) imply in reduced short range marginal costs and, in consequence, low spot prices, because these situations there is no need of thermal plant operation. On the other hand, periods with reduced inflows (dry period) imply high spot prices. Therefore, the financial risks are quite high due to the present and future hydrology dependence.

To establish long term bilateral contracts, an agent's decision should be made suitable, minimizing the effects of the exposure to spot prices volatility. The agent also needs to consider the following parameters: System operating conditions; generation system investment decisions; generating system evolution, given by the PX auctions; and the insights of each agent regarding market projections.

3 Multi-agent Systems

An agent is a physical or virtual entity which is capable of acting in an environment, and it can communicate directly with other agents. It is guided by its purposes and it

is capable of perceiving its environment. An agent has abilities and can offer services despite having only a partial representation of its environment (It is also possible that an agent has no representation at all). Multi-agent systems (MAS), according to [6], are systems with the following elements:

- An environment;
- A set of objects;
- A set of agents;
- An assembly of relations which link objects to each other;
- An assembly of possible operations to agents.

MAS belong to a branch of Artificial Intelligence (AI) related to the study and development of distributed and parallel solutions, called Distributed Artificial Intelligence (DAI). In this manner, MAS inherit several motivations of DAI, e. g., modularity, parallelism and flexibility [6].

Communication between agents permits information exchange in a multi-agent environment, allowing cooperative and coordinated agent actions. The communication pattern is the Agent Communication Language (ACL) defined by the Foundation for Intelligent Physical Agent (FIPA) [7]. FIPA identifies the key roles of the agents required for plataform management and specifies the management agent language content. There are several key functions in this architecture: AMS, DF, MTS and ACC.

The Agent Management System (AMS) agent is a supervisor, controlling uses and acess to the plataform. It authenticates resident agents and controls its registration. The Directory Facilitator (DF) agent provides the agent name to plataform. The Agent Communicator Channel (ACC) provides basic communication between agents inside and outside plataform. The Message transport system (MTS) agent´s function is the exchange of messages between agents.

4 Proposed Model

In this paper, we are concerned with one main challenge: to use MAS technology in an operational and commercial model of a tight pool wholesale energy market. The MAS is also employed as a set of tools to analyze strategically the behavior of power generators and distribution companies, in a system governed by market rules, and composed by hydroelectric and thermoelectric generators, free consumers and distribution companies. Therefore, the DYNAMIS simulator environment is composed, firstly, by eleven agent types (plus three JADE default communication agents) described below:

- *Study configuration agent (CONF)* – permits simulation parameters configuration through a GUI front-end (See Fig. 1);
- *Independent system operator (ISO)* – simulates the independent system operator behavior. Implemented functions are: inflow forecasting; long term operation planning; hydrothermal scheduling; evaluation of stored energy systems;

commercial dispatch ("ex-ante"); and technical generation dispatch ("ex-post") with system contraints;

- *Power exchange (PX)* – simulates the power exchange and differences tight pool electricity market behavior. Its functions are: register company agent bilateral contracts; carry out the settlement processes to determine "ex-ante" and "ex-post" spot price; apply reallocation energy mechanism rules to hydropower plants; publish reports with settlement results for wholesale market agents in the business layer;
- *Nature agent (NAT)* – represents the natural researches behavior and determines the simulation horizon, the natural inflows scenarios in hydro power plant sites and temperature scenarios in thermal power plant sites;
- *Transmission grid (GRID)* –represents all transmission components such as power lines, substations and equipments. Its functions are: evaluate outage in transmission grid components; evaluate the physical viability of generation dispatch "ex-ante" and "ex-post" set by the ISO agent and load flow calculus;
- *Distribution grid (DGA)* – represents the total load of captive consumers' behavior in each distribution/transmission company border substation. Its functions are load forecasting and determine the load scenarios on simulation horizon;
- *Free consumer (FCA)* – simulates the free consumers' behavior in the BWEM. Its functions are: load forecasting and scenarios on simulation horizon; account and analysis of bilateral contracts costs in cash flows structures approaches; bilateral contract establishment with GCA agents;
- *Thermoelectric generator (TGA)* - models the behavior of thermoelectric generators in the system. Its functions are: calculate the energy production function; evaluate generation unit outages with a discrete event generation based on Monte Carlo method; determine the generation units maintenance program; participate of "ex-ante" and "ex-post" dispatch process; temperature forecasting;
- *Hydroelectric generator (HGA)* - models the behavior of hydroelectric generators in the system. Its functions are: evaluate the water conservation law in power plant dam; calculate the energy production function; evaluate generation unit outages with a discrete event generation based on Monte Carlo method; determine the generation units maintenance program; participate of "ex-ante" and "ex-post" dispatch processes and natural inflows forecasting;
- *Generation company (GCA)* - represents the behavior of generation assessments owners. Its functions are: account and analysis of bilateral contracts revenues and generation costs in cash flow structures approaches; bilateral contract establishment with DGA and FCA agents;
- *Distribution company (DCA)* - represents the behavior of distribution assessments owners and captive consumers (market). Its functions are: account and analysis of bilateral contracts costs in cash flows structures approaches; bilateral contract establishment with GCA agents.

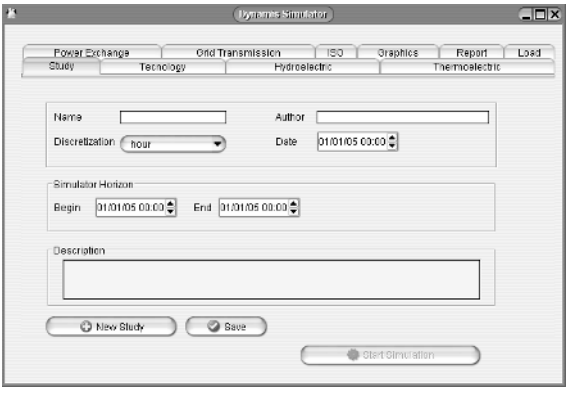


Fig. 1. CONF agent front-end of DYNAMIS simulator

The proposed multi-agents framework is based on a JADE template of management described by FIPA and considers a computing environment developed in the JAVA language. Each agent can run on a different network host. The structure of each agent and their interation with JADE framework is presented in Fig. 2.

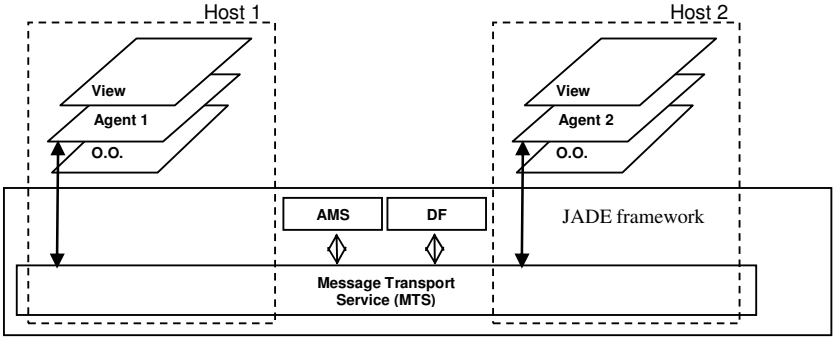


Fig. 2. Individual agent and JADE framework structure of DYNAMIS simulator environment

In MAS each agent communicates with its neighbours and environment. For power industry MAS simulators, Koritarov (2004) [8] suggests the use of interaction layers, i.e., a multidimensional environment in which agents operate within several interconnected layers, such as a physical layer, several business layers and a regulatory layer. Another approach to the interaction layer is proposed here. The DYNAMIS simulator was developed under a new layer perspective. It was implemented using four layers, described below:

- *Natural physical layer* – This layer that represent real natural resources (e.g. petroleum and natural gas reservoirs), natural phenomena (e. g., temperature, inflows, pressure, humidity, winds, solar radiation) and geographical aspects such as terrain topography, rivers and lakes. The geographical position of objects and

agents is important. Agents of this layer can be characterized by use of databases of measuring stations spatially distributed over the geo-electric control area;

- *Artificial physical layer* – This layer represents the components of power system like power plants, power lines, substations, equipments and load units. Geographical position representation of objects and agents is important. Agents of this layer are characterized by physical modeling of their components and operational constraints;
- *Control layer* – This layer has agents that execute the functions of an energy management system, like dispatch coordinate control (ISO functions). As in competitive electricity markets the "ex-ante" dispatches can be made by a PX operating in this layer. Agents of this layer are characterized by specialized functions with control, forecasting, optimization and rule based methods;
- *Business layer* – In this layer the power industry company agents, like GCAs and DCAs, are implemented. Agents of this layer are characterized by learning methods in cash flow modeling and analysis.

The advantage of a multi-layer interaction approach to agent design is to represent entities as they are in real world, separating its attributes and functions. Another issue is that, for specific method performance analyses, some layers can be (de)activated, e. g., for long term operation planning the business layer is not necessary. Likewise, for assessment of a new power flow method, only two physical layers are necessary. Table 1 presents the distribution of agents by layer in DYNAMIS simulator. Fig. 3 shows a general view of the simulator environment.

Table 1. Distribution of agents by layer in DYNAMIS simulator

LAYER	AGENTS
Natural physical	NAT
Artificial physical	HGA, TGA, FCA, DGA, GRID
Control	PX, ISO
Business	GCA, DCA

For example, when an HGA agent requests information from another agent the answer contains detailed and complete representation of its characteristics. In the implementation model a membership agent object is responsible for the attributes and the operational physical and technical representation of a real agent. In DYNAMIS practically all sent messages in the simulation are objects. Thus, a class set was implemented that represents the internal structure and the available behaviors in the objects that described the messages.

DYNAMIS uses a database extracted of SIPOT (Braziliam Power System Information System) available by ELETROBRAS and power plant and company information collected in web sites and entries text files of official Brazilian Power Exchange (CCEE) software that calculates the weekly spot price (NEWAVE).

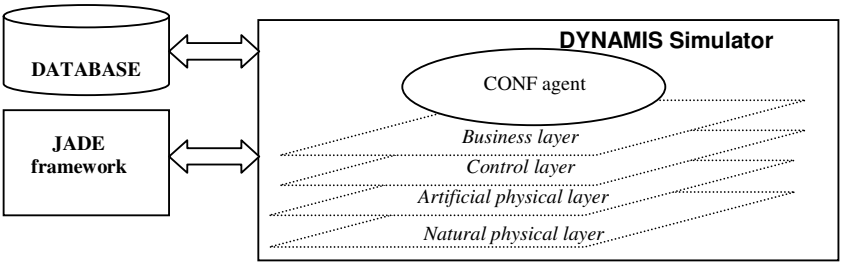


Fig. 3. Global vision of DYNAMIS simulator environment

5 Case Study

A case study was carried out to demonstrate potential applications for the DYNAMIS simulator. A simplified hydrothermal tight pool energy market was simulated with 9,503 MW installed capacity (57.9% hydro, corresponding to 5,502 MW, and 4,001 MW of thermal plants). The total assured energy is equal to 46,612 GWh/year (42.7% hydro and 57.3% thermal). The simulation horizon used was a five year period in monthly intervals. Table 2 present the quantities of artificial physical and business layer agent implemented.

Table 2. Number of agents simulated

Agent	Quantity
DGA	5
FCA	2
HGA	6
TGA	18
GCA	7
DCA	5

The DGA agent contract structure was the same actually used by CELG, CELESC, CPFL and RGE utilities available in electric energy Brazilian public utility commission (ANEEL) technical reports about utilities periodic review tariffary process in 2003-2005 pedriod. The seasonal monthly load of each utility was performed according real utility load as described in Brazilian Association of Electric Distribution Utilities (ABRADEE) web site database of 2000-2004 period. For FCA agents were used as real data based on minning and sewerage real load profile industries.

Control and natural physical layers agents are implemented. The HGA agents implemented are Furnas, Peixoto, Igarapava, Jaguara, Marimbondo and Água Vermelha real hydropower plant of Brazilian hydrothermal system. The TGA agents are the same modeled at NEWAVE software actually used in Brazilian PX and represent 18 thermoelectric power plants of Northeast region. Historic inflows data are used to inflow forecasts and to generate the future inflow cenarious in each hydropower plant. Three operational policies of reservoirs were implemented in the ISO agent “ex-ante” hydrothermal dispatch:

- *Policy 1:* Priorizes the hydro power plant dispatch, i. e., all natural inflows and reservoir storaged energy are used to supply demand. If those are not enough, thermal power plants are dispatched;
- *Policy 2:* This operational politics uses an optimization model that determines the total hydro energy production in the current interval and disaggregates it using non-linear reservoir operation rules (ROR) defined by Soares (1999) [9];
- *Policy 3:* Uses a deterministic optimization model for individual hydro power plants, developed by Soares (1999) [9].

The figures below present the impact of three different operational politics. It is clear that system storage energy is preserved using optimization methods, while preserving system reliability and low average spot prices on the spot market. Many methodologies and analyses with the flexible and multi-platform DYNAMIS simulator can be carried out.

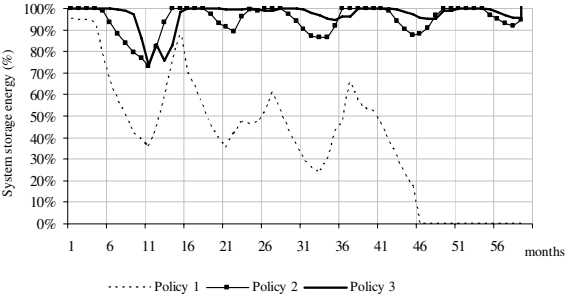


Fig. 4. System storage energy trajectories impacts of different operational policies

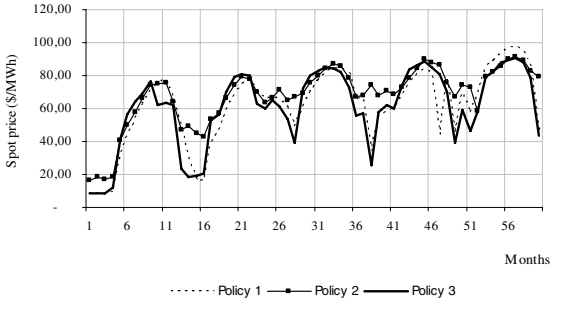


Fig. 5. System spot price impacts of different operational policies

6 Conclusions

A Brazilian wholesale energy market simulator based on multi-agent technology was proposed. The software tool was carried out using multi-layer concept to define specific procedures and objective of software agents. The JADE framework was used to manage agents. Fourteen types of agents were defined and implemented. Two

approaches to multi agent systems applied to power markets industry analysis were developed: operational and wholesale market modes.

The simulator is a powerful tool for Standard Market Design analysis, to define strategic market analyses, and for testing optimization tools in the power industry as a whole, leveraging the flexibility, distributed and parallel processing aspects of multi-agent systems.

Many learning machine, intelligent systems, forecasters, optimization and heuristics models can be evaluated in the proposed simulation model. Variations of tight pool wholesale energy market aspects can be analyzed with this integrated operational and commercial framework tool.

Loose pool models with several auction structures can also be developed and integrated into the PX agent. The event sequence is easily modified, so that the commercial dispatch ("ex-ante") is released on PX agent, instead of being carried out in ISO agent. The remaining simulation loop events may be similarly modified. The proposed simulator is a powerful tool for analysis of various market structures, demonstrating the reuse capacity of classes end functions developed.

References

1. Liu, H., Yuan, B., Dai, H., Lin, J., Ni, Y. X., Wu, F. F.: Framework Design of a General-purpose Power Market Simulator Based on Multi-agent Technology. IEEE Power Engineering Society Summer Meeting, 2001, volume 3, 15-19 july (2001) 1478-1482
2. Praca, I., Ramos, C., Vale, Z.; Cordeiro, M.: MASCEM: a multiagent system that simulates competitive electricity markets. IEEE Intelligent Systems, Volume 18, Issue 6, Nov-Dec (2003) 54–60
3. Ferreira, W. D., Vinhal, C. D.N.: A fipa multi-agent framework for hydrothermal electricity generation market analysis. 5th IEEE Induscon Congress. Salvador, Brazil (2002)
4. Carvalho, F. C., Vinhal, C. D. N., Lima, W. S., Ferreira, W. D.: Reinforcement learning multi-agent model for wholesale electricity market analysis. In: 9th SEPOPE – Proceedings of Symposium of Specialists in Electric Operational and Expansion Planning, Rio de Janeiro, Brazil. (2004)
5. JADE. Java Agent DEvelopment Framework. Available: <www.sharon.cselt.it/jade/>. Accessed: (set/2003)
6. Weiss, G.: Multiagent Systems. Cambridge, EUA: MIT Press (1999)
7. FIPA: Pattern communication specification. Foundation for Intelligent Physical Agent, 2000. Available in: <www.fipa.org/specs/aclspecs.tar.gz>. Accessed: (nov/2002.)
8. Koritarov, V. S.: Real-world Market Representation with Agents. IEEE Power & Energy Magazine. Volume 2, Issue 4, July-Aug. (2004) 39-46
9. Soares, S., Carneiro, A. A. F. M.: Reservoir Operation Rules for Hydroelectric Power System Optimization. In Proceedings of the 1993 Athens Power Tech - IEEE/NTVA, (1993) 965-969

Using IDEF0 to Enhance Functional Analysis in $\mathcal{M}$OISE$^+$ Organizational Modeling

André Luiz Pimentel Uruguay[1] and Celso Massaki Hirata[2]

[1] Instituto de Estudos Avançados
Rod. dos Tamoios, km 5,5
12.228-001 São José dos Campos-SP, Brazil
auruguay@ieav.cta.br
[2] Instituto Tecnológico de Aeronáutica
Pça. Mal. Eduardo Gomes, 50 - Vila das Acácias
12.228-900 São José dos Campos-SP, Brazil
hirata@ita.br
http://www.comp.ita.br/~hirata/

Abstract. A proposal to use the Integrated Definition for Function Modeling (IDEF0) as a method for functional analysis in $\mathcal{M}$OISE$^+$ representation is presented. $\mathcal{M}$OISE$^+$ is composed of three types of specification: functional, structural and deontic. It is shown that IDEF0 can capture many context details during functional phase which facilitates the elicitation of initial beliefs, goals and plans for further cognitive agent design. IDEF0 also makes clear distinction between functional and process modeling (defined in IDEF3), leaving to the agent to choose the best process to try to perform a function. A case study involving the organizational modeling of an integrated air defense system with its command and control infrastructure is presented. The assessment of the case study shows how the visibility of inputs, outputs, controls and mechanisms can lead to an easier design of the agents. A multiagent-based simulation from the resulting organizational model is implemented using Belief-Desire-Intention (BDI) cognitive architecture in JADEX Agent System.

1 Introduction

The increased adoption of computers in the management world has prompted the need for the design of novel organizational forms to obtain more benefits from information technology. The fields of Distributed Artificial Intelligence and Computational Organization Theory brought usefull insights to develop ideas for new structures and processes, but in the face of a lack of analytic body to study these alternative organizations, agent-based simulation appears as the most viable technique.

On the other hand, due to cost, time and safety concerns, many application domains demand executable models where both the systems of an organization and the people who controls them are simulated entities. These are called "constructive" simulations, instead of "virtual", where real human beings are "in the

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 78–87, 2006.

loop". Command and control is such a domain, with organizational issues and distributed cognition being jointly addressed [1].

Until recently most of agent-oriented modeling methodologies did not make explicit the concept of organization [2], and those which did it did not distinguish between organizational functions and its structures. This distinction is conceptually important to provide correlation between functions and components of the system [3], making diagnosis and reorganization easier. Although some very rigorous formalisms such as SDML [4] and ODML [5] were created to aid in the organizational design problem, distinct functional descriptions remained absent. Furthermore, for the agent-group-role (AGR) paradigm [6] there were proposals [7] to define a role state as related with its inputs and outputs, in a clear reference to a functional view.

Another issue with methodologies is the lack of attention to real organizations. Some agent-oriented (AO) methodologies are so beared to engineering software applications that focus on organizational aspects exists only to help to elicitate system requirements. Symbolic of this is the adoption of "use cases" [8]. This is worthless when trying to solve complex organizational problems via simulation. In some models agents cannot be taken as "candidates" to meet funcional requirements prescribed in use cases, but they must reflect real entities (e.g. individuals, other organizations or systems, etc.) with real complex problems whose degree of detail in the model is a real concern for the simulation purpose.

If colaborative and specialized work in an organization is to be modeled, then one has to specify a clear, consistent and complete functional description. In this sense the $\mathcal{M}$OISE$^+$ [9] organizational representation appears as an adequate baseline. The present proposal is to provide a process for functional modeling of organizations, virtual or real, which can enable an easier design of a cognitive multiagent system. In section 2 we present a brief background of $\mathcal{M}$OISE$^+$ and IDEF0. We show, in section 3, how the additional information provided by an IDEF0 model can support a further design of a cognitive multiagent system. This is made by using a case study of the modeling of an integrated air defense system. Section 4 presents the conclusions and final comments.

2 Background

2.1 The $\mathcal{M}$OISE$^+$ Organisational Representation

$\mathcal{M}$OISE$^+$ is a proposal for organizational modeling, composed of 3 perspectives: functional, structural and deontic. The functional specification represents a hierarchic decomposition of global goals in subgoals and the grouping of goals in missions. The structural specification expresses the existing roles in an organization and how they are joined in groups. It also represents relations of authority, communications, acquaintance and compatibility. The deontic view relates each role described in the structural model to the goals it can commit to in the functional specification, and the level of this commitment. All these views composes the *Organizational Specification (OS)*. An *Organizational Entity*

(OE) is instantiated when agents adopt roles in the structural specification, consequently taking part in groups and commiting to goals and plans, . For the sake of brevity these perspectives are here briefly described. A detailed explanation can be found in [10]. Also, middleware called $\mathcal{S}-\mathcal{M}$OISE$^+$ [11] was developed for declarative programming of an organization, independent of agents' architecture or implementation technology.

Functional Specification (FS). Functions are depicted as *global goals*. Being $\mathcal{G}$ the set of global goals, each global goal $g \in \mathcal{G}$ can be further decomposed in *subgoals* g_i or plans and assigned to agents in the form of *missions*, in a tree-like scheme. The leaves of this tree are atomic goals which an agent can try to accomplish. Missions are expressions containing goals and some operators. The operators included resemble concurrency issues: sequence (","), choice ("|") and parallelism ("||"). An example of functional specification is presented in the case study of section 3, Figure 2.b, where, for instance, mission m_1 would be $m_1 =\{$*detect incursor aircraft,alert fighter,direct interception*$\}$. Hereafter this method will be refered to as the "baseline".

Goals can also have special attributes such as: (i)degree of success: indicates if the goal was satisfied or if it is impossible to meet; (ii)alocation level: indicates if any agent is already commited to the goal; and (iii)activation level: indicates if pre-conditions to accomplish the goal are satisfied.

Structural Specification (SS). This specification shows sets of behaviours an agent must comply when it adopts a role (the individual level); the acquaintance, communication and authority relations betwenn roles (the collective level); and the aggregation of roles in groups (the social level). Figure2.a depicts the roles and its structure in the case study. Individual agents are associated with one or more roles within the organization. *SS* is a template where one can eventually find roles which are suitable for one or more agents' capabilities, enabling them to make part of the organization.

Deontic Specification (DS). This description aims to state the concretization parameters to realize the organization. In the $\mathcal{M}$OISE$^+$ framework this is made by assigning permissions and obligations to each role to try to accomplish missions of FS. For instance, an agent performing the role *attacker* could have the obligation to commit to mission m_3.

2.2 The IDEF Methods

Based on structured analysis and design, a suite of methods called IDEF (Integrated DEFinition) was proposed to support analysis of functions and activities and capture of "as is" processes within an enterprise organization. The first completed methods were named IDEF0 (for functions), IDEF1 (for information) and IDEF2 (for dynamic behaviour). Currently IDEF0 and IDEF1X (an extension of IDEF1 for data) are active IEEE Standards and, with IDEF3 (for processes), have broad use in systems engineering.

2.3 Overview of the IDEF0 Functional Model

An IDEF0 model is a structured set of diagrams with boxes and arrows used to specify function models in a very high level. As shown in Fig.1.a, a function is expressed as a rectangle called "ICOM" box containing inputs (I), controls (C), outputs (O) and mechanisms (M). Arrows do not represent flow or sequence as in traditional process flow models. Arrows convey data or objects related to functions to be performed. The functions receiving data or objects are constrained by the data or objects made available.

Each side of the box has a standard meaning: arrows entering the left side of the box are inputs. Inputs are transformed or consumed by the function to produce outputs. Arrows entering the box on the top are controls. Controls specify the conditions required for the function to produce correct outputs. Arrows leaving a box on the right side are outputs. Outputs are the data or objects produced by the function. Arrows connected to the bottom side of the box represent mechanisms. Upward pointing arrows identify some of the means or resources that support the execution of the function.

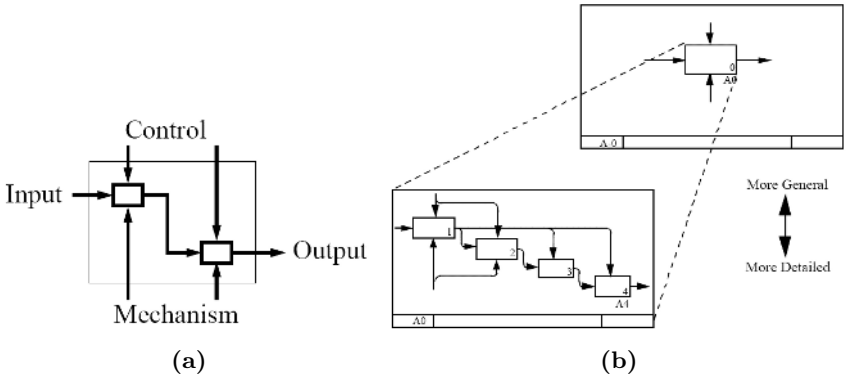


Fig. 1. IDEF0 Notation:(a)Function Box(b)Diagram Decomposition

A special diagram, called *A-0* (reads "A minus zero"), expresses the global function of the system, usually called *A0*. Each box can be decomposed in new subdiagrams in a hierarchic manner. Fig.1.b shows box A0 which belongs to *A-0* diagram. *A0* is decomposed in a new subdiagram with four subfunctions, where inputs, outputs, controls and mechanisms are linked either internally or to other functions in the parent diagram. This can be made repeatedly as deep as needed for the desired degree of detail of the model. There can be other "root" diagrams distinct from A-0 (A-1, A-2, etc.) for different global contexts, but each of them must contain only 1 box, meaning the single global function of the organization.

Finally, IDEF0 is not just a structured way of drawing functions. It has its ground on logic, and a formal description of its syntax and semantics can be found in [12].

2.4 Comparison of Features

Obviously both methods (IDEF0 and the baseline) have similarities. In both cases global functions are decomposed in a structured, hierarchic way: the baseline, by using a goal tree. IDEF0 uses nested diagrams (as a matter of fact IDEF0 also has a "node tree" diagram to show how diagrams are linked, although its use is not mandatory).

Although both methods convey some information about dependency between functions, the baseline relates goals by applying concurrency ordering operators (sequence, choice, parallelism). The missions assigned to each agents are sets of goals with these same operators. The result of this decomposition is a procedural view on how the global goals should be pursued.

IDEF0 pays no attention to the processes to perform functions, so there are no concerns about concurrency and time. Also the concept of missions is absent. Another method, IDEF3, is tailored for process modeling, and it is out of the scope of this work. In IDEF0 models dependence is expressed by inputs, controls and mechanisms arriving at a function's box. Also, there is no indication of preference or priority in performing a function.

3 Elicitation of Intentions and Context for BDI Agent Design

As case study to apply our process, a simple and notional integrated air defense system (IADS) is modeled. An IADS is usually composed of radar sites, communications (COMMS) stations, air bases, fighters and an interception control center. The detected targets are passed, via COMMS station, to the control center which, in case of an enemy aircraft, commands an interception. This order is passed to the airbase commander, which retransmits it to a fighter, for immediate take-off. After take-off, the fighter reports airborne to the interception control center. Based on the dynamics of fighter and target, reported by the radar sites, the center sends to the fighter the optimal heading to intercept. Once the fighter gets the target in range, it launches its weapons. The resulting status of the target is, then, passed to the center, and the fighter returns to base.

To model virtual enterprises and cognitive simulation, Belief-Desire-Intention (BDI) [13] is the abstract architecture of choice, due to its flexibility to environmental changes, provided by desires, and the reasoning ability expressed by intentions [14].

There are many methodologies suited for BDI agents design.Prometheus [15] deals with BDI concepts throughout the whole development cycle. It covers all phases of an agent-oriented development process [16] and it is well suited [17] to JADEX, the BDI tool used at the time this case study began. JADEX [18] was adopted because of its debugging, logging and tracing features. Prometheus has 3 phases: system specification, architectural design and detailed design.

At this point it is worth mentioning a brief remark about use cases: although object-oriented (OO) community has elected UML [19] as the *de facto* modeling

Table 1. Correlation Between IDEF0, OO and AO (adapted from [22])

IDEF	Use Case (OO)	Agent (AO)
Function	Use Case	Goal
Input	Actor Input	Belief
Control	Conditional Use Case	Context or Invocation Condition
Output	Output to Actor	Action
Mechanism	Actor	Agent Collaboration

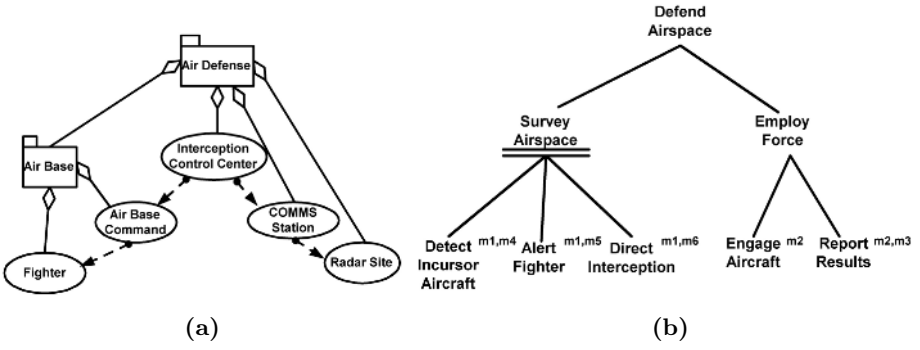


Fig. 2. Air Defense Organizational Specification:(a)Structural(b)Functional

language and there is a possible agent extension [20], the management community still prefers IDEF, for its simplicity and quick learning. UML tends to be, for managers, too abstract and its graphical notation too complex [21]. Kendall [22] points many similarities between IDEF and use case diagrams. Use case extensions, for instance, can be considered as decompositions of IDEF0 diagrams. Table 1 presents a rough correlation between some IDEF0, OO and AO concepts.

Figure 2 presents a $\mathcal{M}$OISE$^+$ structural specification for this case (a), and a possible functional specification by the baseline method (b).

Fig.3 shows an IDEF0 function model for the case at hand. It can be noted the same atomic functions as in the baseline. The small diagram at left is A-0. Box *A0* expresses the root global function,e.g.,"Defend Airspace". A subdiagram at its right, also called *A0*, decomposes it into subfunctions and its ICOM links.

The first step of the process is to examine mechanisms (arrows entering upward IDEF0 boxes). Mechanisms offer rough suggestions on which sets of roles have to be assigned to that function, e.g., "Airbases and Fighters" being mechanisms of "A02-Alert Fighter" functions. Table 2 shows the functions (and their boxes) assigned to each role, the same linkage describe in $\mathcal{M}$OISE$^+$ deontic specification.

Considering the first phase of Prometheus (system specifications) expressed by the organizational model, now it is possible to provide an initial set of goals, beliefs and contexts for a cognitive agent design. The next step is to elicitate the *goals* of the agents, which comes directly from the functions in IDEF0 and roles in Table 2. Each function can have inputs, outputs, mechanisms and controls.

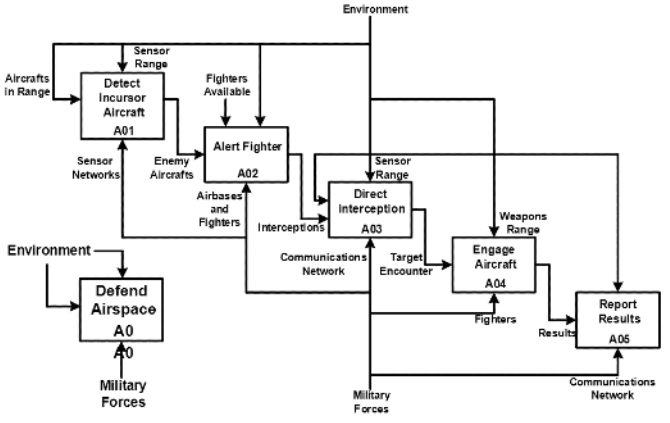


Fig. 3. IDEF0 Functional Specification

Table 2. Functions Assigned to each Role

Role	Function	Box
Interception Control Center	Detect Incursor Aircraft	A01
	Alert Fighter	A02
	Direct Interception	A03
Airbase Command	Alert Fighter	A02
Fighter	Engage Aircraft	A04
COMMS Station	Detect Incursor Aircraft	A01
	Alert Fighter	A02
	Direct Interception	A03
	Report Results	A05
Radar Station	Detect Incursor Aircraft	A01
	Direct Interception	A03

Considering inputs and outputs linked to functions assigned to other roles, these dependences are good candidates for *protocols*. For example, "Alert Fighter" dependence on the enemy aircraft's positions from "Detect Incursor Aircraft" (assigned to a different set of roles) suggests that a protocol must exist between the respective sets of agents. Another set of protocols can be derived from the roles performing the same function, e.g., "Interception Control Center", "COMMS Station" and "Radar Site" share the same function, "Detect Incursor Aircraft".

Another helpfull step is to take inputs to assess which belief sets that each role must possess, e.g., "Direct Interception" most likely demands to know which "Interceptions" are currently in course.

After these steps, a proposed system overview of the architecture can be designed, as shown in Figure 4.

Fig.5 shows the initial detailed design of the fighter agent. In this example each agent is performing one role, described by the structural specification, although in general many roles can be performed by each agent. The detailed model of a

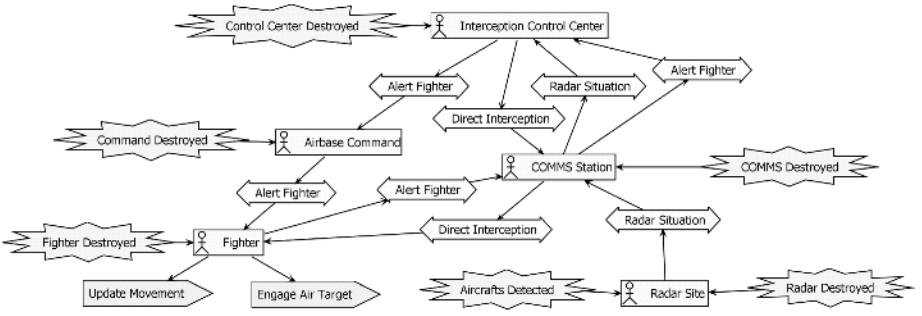


Fig. 4. IADS System Overview

BDI agent can start by assigning a goal for each function assigned to its role, as shown in Table 2.

It is supposed that BDI agents are able to have many plans for each goal and select the best upon context. In this simple case, the single plan has the same name as its goal ("Engage Aircraft"). Again, from the IDEF0 diagram it is possible to extract relevant events, e.g., "Target in Range" suggests an important context conditions for the triggering of the goal "Engage Aircraft".

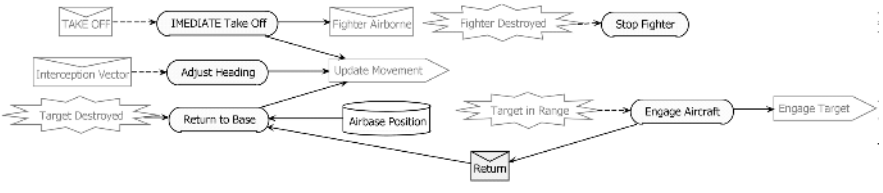


Fig. 5. Agent Overview of Fighter

4 Conclusions and Remarks

Simulation studies are the main source of knowledge for some complex and adaptive domains where an enterprise organization is expected to operate. But in order to capture organizational effects, agile methodologies are necessary. We presented here a proposal for a process to model functions in an organization, free of process details and independent of structure and agent architecture.

IDEF0 has some similarities with the original functional analysis of $\mathcal{M}$OISE$^+$, taken as baseline, but the baseline method expresses some informations that IDEF0 do not capture,e.g., levels of satisfiability, allocation and activation. Also, the concept of mission in $\mathcal{M}$OISE$^+$ is associated with procedures, involving workflow and concurrency operators. In the IDEF0 paradigm, functions are related

to each other by links of data and objects dependency between their inputs, outputs, controls and mechanisms (ICOM).

Processes, in turn, are best described using IDEF3, another method of the IDEF suite. Possibly the joint use of IDEF0 and IDEF3 would enable a representational power that could encompass all the baseline features. Also, the existence of commercial IDEF tools suggests that it can be possible to extend $\mathcal{S} - \mathcal{M}\text{OISE}^+$ to include IDEF0 functional model programming.

References

1. Meirina, C., Levchuk, G.M., Ruan, S., Pattipati, K.R., Popp, R.L.: Normative Framework and Computational Models for Simulating and Assessing Command and Control Processes. Simulation Modelling Practice and Theory **14**(4) (2006) 454–479
2. Kumar, M.: Contrast and Comparison of Five Major Agent Oriented Software Engineering (AOSE) Methodologies (2002)
3. Cummins, R.: "How Does It Work?" versus "What Are the Laws?" - Two Conceptions of Psychological Explanation. In: Explanation and Cognition. MIT Press (2000) 117–144
4. Moss, S., Gaylard, H., Wallis, S., Edmonds, B.: SDML: A Multi-Agent Language for Organizational Modelling. Computational & Mathematical Organization Theory **4**(1) (1998) 43–69
5. Horling, B., Lesser, V.: Quantitative Organizational Models for Large-Scale Agent Systems. In: Proceedings of the International Workshop on Massively Multi-Agent Systems. (2004) 297–312
6. Ferber, J., Gutknecht, O.: A Meta-Model for the Analysis and Design of Organisations in Multi-Agent Systems. In: Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS 98), IEEE Computer Society Press (1998) 128–135
7. Jonker, C., Treur, J.: Relating Structure and Dynamics in Organization Models. In Sichman, J., Bousquet, F., Davidson, P., eds.: Multi-Agent-Based Simulation II, Proceedings of the Third International Workshop on Multi-Agent Based Simulation, MABS'02. (2002)
8. Jo, C.H., Einhorn, J.M.: A BDI Agent-Based Software Process. Journal of Object Technology **4**(9) (2005) 101–121
9. Hübner, J.F., Sichman, J.S., Boissier, O.: Using the MOISE+ for a Cooperative Framework of MAS Reorganisation. In Bazzan, A.L.C., Labidi, S., eds.: Proceedings of the 17th Brazilian Symposium on Artificial Intelligence (SBIA'04). Volume 3171 of LNAI., Berlin, Springer-Verlag (2004) 506–515
10. Hübner, J.F.: Um Modelo de Reorganização de Sistemas Multiagentes. PhD thesis, Escola Politécnica da Universidade de São Paulo, São Paulo (2003)
11. Hübner, J.F., Sichman, J.S., Boissier, O.: S-MOISE+: A Middleware for Developing Organised Multi-Agent Systems. In Boissier, O., Dignum, V., Matson, E., Sichman, J.S.a., eds.: Proceedings of the International Workshop on Organizations in Multi-Agent Systems, from Organizations to Organization Oriented Programming in MAS (OOOP'2005). (2005)
12. USA: IEEE STD 1320.1-1998 - Standard for Functional Modeling Language - Syntax and Semantics for IDEF0. Institute of Electrical and Electronic Engineers, New Jersey (1998)

13. Rao, A., Georgeff, M.: BDI Agents: From Theory to Practice. In: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA (1995)
14. Rahwan, I., Kowalczyk, R., Yang, Y.: Virtual Enterprise Design - BDI Agents vs. Objects. Lectures Notes in Computer Science **2112** (2000) 147–157
15. Padgham, L., Winikoff, M.: Prometheus: A Pragmatic Methodology for Engineering Intelligent Agents. In: Proceedings of the Workshop on Agent-Oriented Methodologies at OOPSLA 2002. (2002)
16. Cernuzzi, L., Cossentino, M., Zambonelli, F.: Process Models for Agent-Based Development. Journal of Engineering Applications of Artificial Intelligence **18**(2) (2005)
17. Sudeikat, J., Braubach, Lars andPokahr, A., Lamersdor, W.: Evaluation of Agent-Oriented Software Methodologies - Examination of the Gap Between Modeling and Platform. In: Lecture Notes in Computer Science. Volume 3382., Springer-Verlag (2005)
18. Pokahr, A., Braubach, L., Lamersdorf, W.: Jadex: Implementing a bdi-infrastructure for jade agents. EXP - in search of innovation (Special Issue on JADE) **3**(3) (2003) 76–85
19. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley (1998)
20. Odell, J., Parunak, H.V.D., Bauer, B.: Extending UML for Agents. In Wagner, G., Lesperance, Y., Yu, E., eds.: Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National Cconference on Artificial Intelligence, Austin,TX (2000) 3–17
21. Cochran, L., Wheaton, K.: A Conceptual Operational Model for Command and Control of International Missions in the Canadian Forces. In DoD Command and Control Research Program, ed.: 7th International Command and Control Research and Technology Symposium, Qubec (2002)
22. Kendall, E.A., Malkoun, M.T., Jiang, C.: A Methodology for Developing Agent Based Systems for Enterprise Integration. In: Proceedings of the First Australian Workshop on DAI. (1996)

Simulations Show That Shame Drives Social Cohesion

Klaus Jaffe

Universidad Simón Bolivar
Apartado 89000, Caracas 1080, Venezuela
kjaffe@usb.ve

Abstract. Using the agent based simulation model Sociodynamica, I explore the effect of various intensities of feelings of shame in cementing social bonds that affect the aggregate health (average longevity of agents) and wealth (GDP) of two artificial societies: A primitive society with no division of labor, and a monetized society with division of labor for three different tasks. The results show that agents who feel shame are successful in colonizing virtual societies, but the effect of shame in achieving cooperation differs between both societies studied They results also show that small levels of shame are better in enhancing social cohesion than large costs imposed with altruistic punishment.

1 Introduction

Aristocles of Athens, better known as Plato (427-347 AC) and Protagoras of Abdera (c. 490 - c. 420 AC) recognized that feelings of shame are fundamental for the maintenance of social cohesion. We define shame as the instinct or innate drive that evokes a feeling of pain if one's behavior is disliked or rejected by others. Shame or equivalent social devices are detectable also among many animals, such as dogs, apes [36] and others. Shame is enhanced and fomented by modern and ancient cultures although exceptions to its usefulness for achieving social cohesion are known [25]. Its social power is used in the implementation of "bench marking" as practiced by business and management specialists, by using shame as a driving force for modifying behavior. As emotions can be simulated with artificial life [13], here we tried to make operational the concept 'shame' for simulation purpose. Operationalization of a emotion might be performed in many different ways [21], here we operationalize 'shame' by focusing on its social function, so that higher levels of shame increase the intensity of compliance with the social norm that elicited that feeling of shame. The aim of the exercise was to create a metaphor of a society that allows us to gain an increased understanding of the role of emotions in dynamic social interactions.

One way to assess the effect of shame is in the context of reciprocity. Reciprocity is theoretically unlikely to evolve as a result of natural selection. Yet such behaviors evidently exist, and not only among humans [35]. No consensual explanation as to the forces allowing for its existence and maintenance exist [19, 31].

Thus human and animal cooperation and altruism continues to remain a puzzle. Among the most recent explanations proposed for solving this puzzle we might cite the following: Altruistic norms can 'hitchhike' on the general tendency of internal

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 88 – 97, 2006.

norms to be personally fitness-enhancing [32] and that a multi-level selection, gene-culture co-evolution argument then explains why individually fitness-reducing internal norms are likely to be pro-social as opposed to socially harmful [10].

Alternatively, neutral non-social players might stabilize the system enhancing the chances of altruistic behaviors to settle in social populations [12]. Another argument is based on the observation that reputation may foster social behavior among selfish agents, and is considerably more effective with punishment than with reward [31]. Previous studies have shown that behaviors that benefit others at one's own expense (altruism) are evolutionary stable (i.e. reach a Nash equilibrium) if many interactions are with genetically related [11] or with genetically similar individuals [14], so that kin selection or homophily can work. Also, repeated interactions allow for the recovery of the altruistic expense [1, 3, 23, for example], converting the act into social investment [15]. Simulations have been very important in this endeavor and interestingly, they suggest that societies allow for the occurrence of synergies that provide hidden or retarded benefits to the various actors, allowing natural selection to maintain behaviors that cement social bonding, such as altruism [14,15].

Evidence for long term hidden benefits of apparently altruistic behavior has been reported in the context of religions [2]. A more recent argument put forward to explain the maintenance of altruistic and of reciprocal behaviors is altruistic punishment. Support that the punishment of non-cooperators at a cost to the punisher and the punished may favor adaptation of altruistic behavior has been provided by field studies [9]. The agent based simulation model Sociodynamica confirmed that altruistic punishment is evolutionary stable, but due to its effect on the individual and society, should be rather viewed as a social investment [17].

Social simulations are one way to disentangle complex concepts. Such models have been used to study the effect of shame, although for different purposes, such as simulating "digital blush" [28]. The agent based simulation Sociodynamica [15,16] provides a framework that allows testing the robustness of social, psychological, biological and economic concepts. This model was applied to explore "altruistic punishment". Altruistic punishment means that individuals punish transgressors of fair play, although the punishment is costly for them and yields no direct material gain. Fehr & Gachter [9] proposed that altruistic punishment might explain the fact that people frequently cooperate with genetically unrelated strangers, often in large groups, with people they will never meet again, and when reputation gains are small or absent. These patterns of cooperation cannot be explained by the nepotistic motives associated with the evolutionary theory of kin selection and the selfish motives associated with signaling theory or the theory of reciprocal altruism. Fehr & Gachter [9] showed experimentally that the altruistic punishment of defectors is a key motive for the explanation of cooperation. Using human subjects, they showed that cooperation flourishes if altruistic punishment is possible, and breaks down if it is ruled out. They also showed evidence indicating that negative emotions towards defectors are the proximate mechanism behind altruistic punishment, confirming previous findings [29]. Yet, altruistic punishment or any other kind of social investment cannot work if the individual punished does not respond to punishment. One such punishment is shaming, where short term economic cost may be involved but are not central to its implementation as other costs, such as future social opportunities, become more important. The feelings of the shamed agents may affect its future interactions with other agents.

In this paper I use the agent based simulation model Sociodynamica to explore the usefulness the concept "shame". I explore the effect of various intensities of feelings of shame on the aggregate health of a society with and without a rudimentary division of labor. The concept of shame, as used here, differs from that of image scoring [24] or shunning [26]. Here, the emotion is particular to the punished agent and determines its future behavior, not that of other agents. It is the reaction of the agents after receiving a punishment for not being sufficiently altruistic.

The results of this preliminary exploration of the effect of shame on virtual societies show that even if a reduced proportion of agents have even small levels of this feeling, the social regulation of the society is enhanced enormously.

2 Methods

The agent based computer simulation Sociodynamica was used to study the effect of altruistic punishment on aggregate health in artificial societies, measured as the average longevity of agents. A somewhat simpler version of Sociodynamica was described before (Jaffe 2002a, b). The model simulated a continuous two-dimensional toroidal world (500x400 pixels) through which different types of agents wander with Brownian motion, each at is proper speed. The speed of this motion (*m*) ranges from 0-30 pixels / time step. Agents did not learn. The simulations tested for the survival abilities of agents under variable circumstances. As dead agents were substituted by new ones, which had their parameters assigned at random, the simulations served as a way of weeding out those combinations of parameters that conferred low survival capabilities to agents, selecting those agents possessing parameters that conferred them larger survival possibilities. Agents did not inherit their parameters, as Sociodynamica is a metaphor for a society of agents living in a free competitive market with infinite replacement potential.

The virtual society was programmed to have different levels of interaction. A first level simulated interactions with the environment, where agents collected food and/or minerals. A second level simulated the interchange of goods between agents. A third level simulated agents that could show altruistic behavior by donating excess food to starving agents. A forth level simulated agents that punished stingy (non altruist) agents, with a cost to themselves and to the punished agent. The fifth level of simulated interactions included the feeling of shame in punished individuals, in which shameful agents would increase their altruistic donations after receiving punishment in accordance to their intrinsic level of shame.

Interactions with the environment

The toroidal world was supplied with patches of agricultural land (food resources: *Rf*) and mines (mineral resources: *Rm*). Each time an agent happened to land over one of these resources while walking randomly around, they acquired a single unit (*wo*) of the corresponding resource, accumulating wealth, either as food (*wf*) and/or as mineral wealth (*wm*). The probability of arriving and resting over a patch of food or minerals was dependent on the speed and type of movement of the agent. Similarly, its ability to serve as a trader also depended on these two parameters.

Agents spend some of their wealth in food in order to survive, consuming food at a basal constant rate (*b*), which was a fraction of the resource unit (*wo*). The wealth in food (*wf*) of each agent changed each time step:

$dwf = -b + wo$ where $wo = 0$ if no resources are encountered.

b determined the degree of external constraints or of competitiveness of the environment and was fixed at 0.1, indicating the speed of degradation of accumulated resources in *wo* / time-step. This value produced simulation outcomes that are closed to what we expect in real societies (Jaffe 2002a,b). Agents with no food resources left ($wf = 0$) perished and were substituted by a new agent with randomly assigned parameters. This substitution process allowed maintaining the total number of agents in the population constant.

Similarly, agents encountering minerals acquired a single unit of the resource (wo) each time they encountered it. Minerals never degraded (bm = 0). The wealth in minerals (wm) was inversely related to the probability of suffering a fatal accident for each agent. That is, mineral wealth improved the odds of surviving external "catastrophes" that killed agents at random, each time step, and large amounts of wm protected the agents against these catastrophes by reducing the probability of being affected by them. Agents with wm = 0 could survive, though, with a lower probability. The agents were struck by a fatal catastrophe if the following relation was true:

$$wm < rnd(0\text{-}1) * D$$

So that the greater the wealth of accumulated minerals of the agent, the lower their probability of being struck by a fatal accident or catastrophe, at any level of danger (D). Both types of resources were replenished continuously. Each of them was concentrated in a different single patch and the total amount of resources was 200 *wo* for food and 100 *wo* for minerals. Each resource patch was distributed initially at random in the landscape but remained in the same place during the duration of each run.

Interchange of goods

Agents moved in random directions each time step. Each time an agent met another at a distance smaller than 20 pixels, an exchange of wealth could occur. These could be of two different types. In some simulations, more "structured societies" were simulated by modeling labor specialization of the agents. In this case, agents were subdivided into three categories. Farmers which specialized in collecting only food; miners which collected only minerals; and traders. Traders specialized in trading minerals for food when encountering a farmer, and food for minerals when encountering a miner. These transactions were performed with money which allowed for the establishment of prices for commodities depending on their supply and demand. When not explicitly stated, artificial societies had no money and no division of labor: and all agents could collect food and/or minerals and could trade their goods.

Altruism

Donations of food occurred when the difference in food wealth (*wf1-wf2*) between the two agents was larger than 2. Then the richer agent transferred food to the less wealthy. The amount of food transferred depended on the generosity (*g*) of the donating agent, which varied initially among agents from 0 to 5 deciles of their wealth (*wf*), i.e. 0 to 50 % of their wealth.

As shown with Sociodynamica before, altruism can invade a population of agents if it provides social benefits, i.e. is rather more like a social investment [17]. Here we

simulated that altruistic donation in food benefited society and the individual indirectly by increasing the *wm* of the donor in a percentage of the donation given by the variable *gain*, with a default value of 15%.

Shame

Agents were shamed by agents acting as altruistic punishers in the population. The proportion of shamers (altruistic punishers) was kept constant during each simulation run. The shamer imposed a fixed cost to the shamed agent and had to pay that same cost himself.

The intensity of the reaction of agents when shamed was coded as sensitivity to punishment or degree of the feeling of shame, which regulated the degree to which shamed agents increased their generosity after having been shamed. This increase could vary from 0 to 10 deciles of *wf*.

Simulation runs

The values of parameters not specifically analyzed or described as being fixed in a simulation, were allowed to vary randomly among agents in the ranges mentioned above. Unless stated otherwise, simulations were run 200 times with 1000 agents for 100 time steps. Although the configuration of populations never stabilized completely, after 60 time steps changes were very small. Thus, the populations of agents reached a stable state after time interval of 100 time steps.

3 Results and Discussion

The simulation results showed that the possession of high levels of shame was not a handicap for agents. Figure 1 shows the frequency distribution of agents possessing various levels of shame, after a simulation run of 200 time steps.

Here simulations were run so that agents could adapt their levels for shame from 0 to 10 deciles, showed that they did not extinguish shame. That is, agents which felt no shame (shame = 0) did not displaced agents feeling shame. The figure shows the final distribution of agents (number of agents in the y axis) according to their levels of shame (in deciles of generosity increase after receiving a punishment given in the x axis) after a typical 200 time step run of a simulation of competing agents.

Clearly, shame was adaptive in our model as increased generosity slightly increased the odds of surviving fatal accidents. For poor agents, their level of generosity had no effect on their fitness as their threshold for donating was never triggered.

As programmed in the model, levels of cooperation defined as amount of generosity, correlated with higher survival rates. This correlation, however, was non linear, as shown in the Figure 2. The results presented in Figure2, shows the average outcome of 200 simulations (each dot represents one simulation) after 100 time steps, regarding the levels of generosity (red squares, right axis) and longevity (blue circles, left axis) achieved by agents, with the programmed gain to society altruistic acts could achieve (gain through cooperation), as agents were living in worlds were cooperation could achieve different levels of social gains.

The figure shows a non linear relationship between the ethical variables (levels of generosity) and biological variables (longevity) of agents in relation to the amount of

synergy (gain through cooperation) that the society could achieve through its altruistic behavior. The more society could gain through the altruistic actions of its agents, the more likely agents were altruistic, and the longer the average longevity of agents.

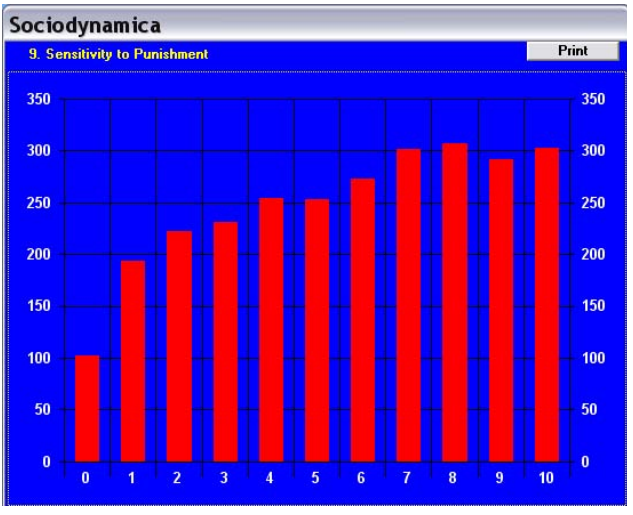


Fig. 1. Distribution of levels of shame (sensitivity to punishment simulated as increased generosity shown after receiving a punishment) after 200 time steps in a virtual population of agents submitted to strong inter-agent competition

In general, the interaction between the various parameters studied and its effect in increasing cooperation was complex and here we present only a selection of the relevant results.

In the next graphs (Figure 3 and 4) present the effect of the cost of altruistic punishment and that of the intensity of shame on the average age of the agents in the population. In the model, less cooperation increased the mortality of agents, thus, longer ages indicates that more successful cooperation was achieved. This effect was studied in primitive societies of barterers and collectors with all agents being generalists (no division of labor).

In these two graphs we see the effect of the cost of the altruistic punishment (paid by both, the punisher and the trader) on the average longevity and generosity of agents. The results show that the higher the cost, the lower the mean generosity of agents and the lower the life expectancy of agents, as lower levels of cooperation was achieved. We also see that a small cost reduces live expectancy significantly, but the effect in reducing the levels of generosity among the agents is much smaller. These effect were evident with all levels of shame explored.

The next two graphs (Figures 5 and 6) show the effect of increasing the degree of shame felt by agents that triggers increasing generosity, on the average longevity and generosity of agents. The results show an interesting non-linear effect. For large costs of altruistic punishment, no cooperation is established which reflects in low average

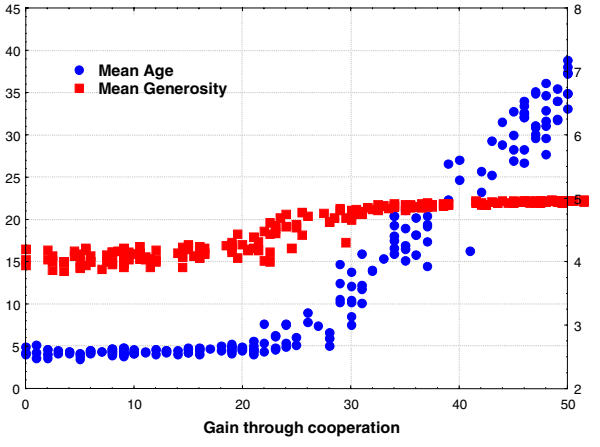


Fig. 2. Result of 200 simulations (each dot represents one simulation) after 100 time steps, regarding the levels of generosity (red squares, right axis) and longevity (blue circles, left axis) achieved by agents

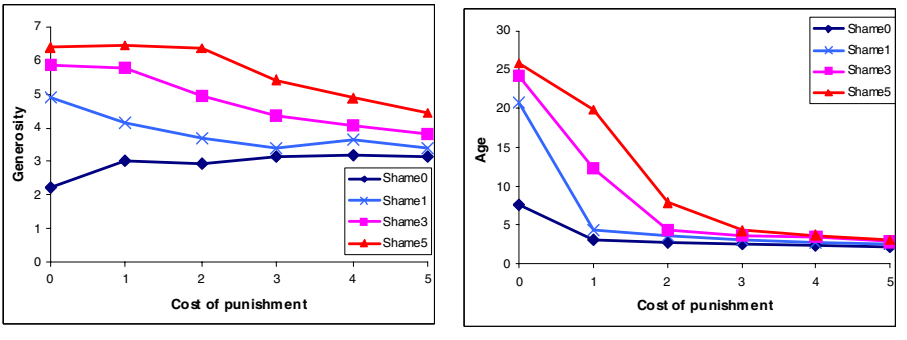
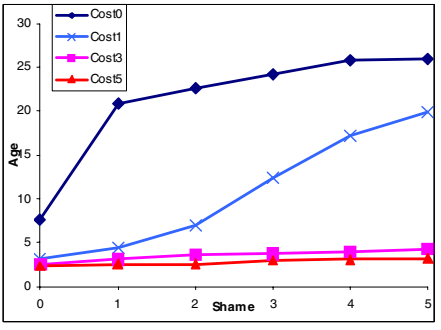


Fig. 3 and 4. Average generosity and age of agents, after 200 time steps, for simulations with different costs of punishment and different fixed levels of shame

longevity, whereas for smaller costs, small levels of shame allow the establishment of cooperation that achieves high longevity. With shame = 0 and costs = 0 we obtain cooperation that is not driven by punishment.

All the simulations described until now were performed on artificial societies that had no money and no division of labor. But Sociodynamica also allows for the simulation of more complex societies. The next graph (Figure 6) shows the results of a simulation on a monetized society where agents were farmers, miners or traders (Division of labor).

Here interesting effects were evidenced. Although in general, the same trends as those described for simple societies were observed, these trends were less conspicuous, and a strange overlap of the curves was observed in the graph at low values of feelings of shame. These results suggest that the effect of feelings of shame on some aggregate variable of a society will depends on the structure and relationship network of that society.



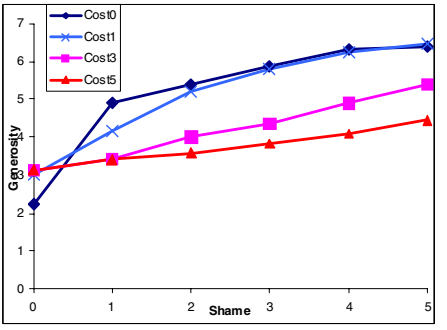


Fig. 5 and 6. Average generosity and age of agents, after 200 time steps, for simulations with different costs of punishment and different fixed levels of shame

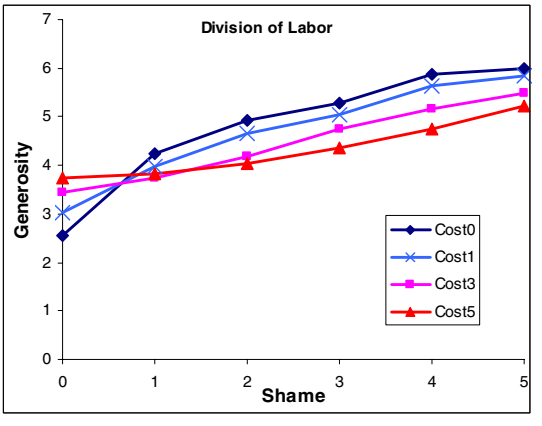


Fig. 6. Relationship between generosity and shame for different costs of punishment in a more structured society

4 Conclusions

The work is based on the assumption that cooperation is a social investment that eventually benefits the average agent, including many of the individuals doing the long term social investing or "altruistic" donation. In our simulations, the pro-social behavior simulated, benefited society as a whole, and being shamed had a cost that was reflected in the act that not being shamed increased the odds of survival in the future by small amounts. Under these conditions we saw an effect of both, the cost of altruistic punishment and the levels of shame of the agents, on the achievement of cooperation (level of generosity) in the virtual society. The interesting point is that both features, the cost of punishment and the reaction to punishment (shame), although part of the same behavior, differed in their effect on the aggregate properties of society. Large levels of shame favor social behavior whereas large costs for altruistic punishments hinder them.

The results clearly show that introducing shame in the virtual society is more effective for the achievement of pro-social behavior than increasing the punishment to non-complying agents.

In terms of evolutionary biology, shame might have been stabilized by evolution for any other purpose, as for example stabilizing family life and/or making education of offspring by parents more efficient. One the genetic and phenotypic tools for triggering shame have become well established, these could be used for other purposes, such as stabilizing societies. It is not the feeling of shame itself that is relevant for the adaptive behavior of a species but its role in triggering behaviors that stabilize or improve social relationships. Thus, shame can be viewed as a kind of long term indirect social investment, that helps the individual to regulates its behavior so as to improve the efficiency of social mechanisms that will eventually also benefit him.

This work shows the relevance of the context in which we analyze emotions. Looking at emotions in the wrong context will not unveil their adaptive function. Emotions that are exclusive to social species have to be analyzed in a social context in order to understand their working. Thus, much simpler models might not unveil features of emotions that are relevant to their social function. A more sophisticated analysis of social contexts might reveal novel functions for emotions such as shame.

Artificial societies and computer simulations, of course, are very distant from reality. The insight gained from this exercise, I suggest, should be that shame, and may be other feelings, do have important consequences on the aggregate properties of a society. The quantitative and qualitative distribution of feelings of shame in a real population should be explored in order to relate them to macro socio-economic variables of the society. This would allow correlating the prevalence and distribution of socially relevant emotions on the working of society. The results of the presented work would suggest that we might find some surprises when researching these questions in real societies.

References

1. Axelrod, R. and Hamilton, W. The evolution of cooperation. Science, 211: 1390–1396, 1981.
2. Berman, E. Hamas, Taliban and the Jewish Underground: An Economist´s View of Radical Religious Militias. NBR Working Paper 10004, 2003 http://www.nber.org/paoers/w10004
3. Boyd, R. and Lorberbaum, J.P.. No pure strategy is evolutionarily stable in the repeated Prisoner's Dilemma game. *Nature* 327, 58 – 59, 1987.
4. Boyd, R. and Richerson, P.J. Punishment allows the evolution of cooperation (or anything else) in sizable groups. Ethology and Sociobiology 13: 171-195, 1992.
5. Boyd, R., Gintis, H., Bowles, S., Richerson, P.J. The evolution of altruistic punishment. Proceedings of the National Academy of Science 100: 3531-3535, 2003.
6. Brosnan, S.F. and de Waal, F.B.M. Monkeys reject unequal pay. Nature 425: 297-299, 2003.
7. Castelfranchi, C. 2000. Engineering Social Order 1. ESA working paper, 2000. http://lia.deis.unibo.it/confs/ESAW00/pdf/ESAW04.pdf
8. Compte, A. 1830-1842. Cours de Philosophie Positive. Hermann Ed, Paris. 1998.
9. Fehr, E and Gächter, S. Altruistic punishment in humans, Nature 415: 137-140, 2002
10. Gintis, H. The hitchhiker's guide to altruism: gene-culture coevolution, and the internalization of norms, Journal of Theoretical Biology 220: 407-418,2003.
11. Hamilton, W. D. The genetic evolution of social behavior. Journal of Theoretical Biology 7: 17-18, 1964.

12. Hauert, Ch., De Monte, S., Hofbauer J. and Sigmund, K. Volunteering as Red Queen Mechanism for Cooperation in Public Goods Game, Science 296, 1129-1132, 2002.
13. Hille, K. Synthesizing Emotional Behavior in a Simple Animated Character. Artificial Life 7: 303-313, 2001.
14. Jaffe, K. On the relative importance of Haplo-Diploidy, Assortative Mating and Social Synergy on the Evolutionary Emergence of Social Behavior. Acta Biotheoretica 49: 29-42, 2001.
15. Jaffe, K. An economic analysis of altruism: Who benefits from altruistic acts? Journal of Artificial Societies and Social Simulations 5: 3, 2002 http://jasss.soc.surrey.ac.uk/5/ 3/3.html
16. Jaffe, K. Monte Carlo exploration of mechanisms for the creation of aggregate wealth. Proceedings of IAREP/SABE Conference, Turku, Finland. 2002. http://atta.labb.usb.ve/Klaus/MonteCarlo%20Explo%20of%20Wealth.htm
17. Jaffe, K. Altruistic punishment or decentralized social investment? Acta Biotheoretica, 52: 155-172, 2004.
18. Jimenez-Alonso W. The role of kin selection theory on the explanation of biological altruism: a critical review. Journal of Comparative Biology 3: 1-14, 1998.
19. Johnson, D.D.P, Stopka, P. and Knights, S. The puzzle of human cooperation. Nature 421, 911–912, 2003.
20. Kaminka, G.A. and Tambe, M. Robust Agent Teams via Socially-Attentive Monitoring. Journal of Artificial Intelligence Rresearch 12: 105-147, 2000.
21. Krebs, D. and Janicki M. Biological foundations of moral norma. In: Psychological Foundations of Culture, M. Schaller and C. Crandal (Eds.), page 1-14.
22. Lincoln, R., Boxshall, G. and Clark, P. A Dictionary of Ecology, Evolution and Systematics. Second edition. Cambridge University Press. 361pp. 2001
23. Maynard Smith J. Group selection and kin selection. Nature 201: 1145-1147, 1964
24. Nowak, M.A. and Sigmund, K. Evolution of indirect reciprocity by image scoring. Nature 393: 573, 1998.
25. Orr S.W. The Economics of Shame in Work Groups: How Mutual Monitoring Can Decrease Cooperation in Teams. Kyklos, Volume 54, Number 1, 2001, pp. 49-66(18).
26. Panchanathan, K. and Boyd, R. Indirect reciprocity can stabilize cooperation without the second-order free rider problem Nature 432, 499-502, 2004
27. Pearce, D.W. (ed). The MIT Dictionary of Modern Economics. Forth edition. MIT Press, Cambridge Massachusetts. 474 pp. 1996.
28. Pitt, J. Digital blush: towards shame and embarrassment in multi-agent information trading applications. Cognition, Technology & Work. Springer London 6: 23 – 36, 2004.
29. Rawlings, E.I. Witnessing harm to other: a reassessment of the role of guilt in altruistic behavior. Journal of Personality and Social Psychology 10: 377-80, 1968.
30. Chimps Lack Charitable Nature. Nature 2005
31. Sigmund, K., Hauert, Ch. and Nowak, M. A. Reward and punishment, Proceedings of the National Academy of Science 98, 10757-10762, 2001.
32. Simon, H. A Mechanism for Social Selection and Successful Altruism, Science 250: 1665–1668, 1990.
33. Smith, A. An Inquiry about the Nature and Causes of the Wealth of Nations. The Library of Economics and Liberty. 1776
34. Sober, E. and Wilson, D.S. Unto Others: The Evolution and Psychology of Unselfish Behavior. Harvard University Press, 700pp. 1999.
35. Wilson, E.O. Sociobiology: A New Synthesis, Harvard University Press. 1976.
36. Whiten, A. Horner, V. and de Waal, F.B.M. Conformity to cultural norms of tool use in chimpanzees. Nature 423: 737-740, 2005.

SILENT AGENTS: From Observation to Tacit Communication

Cristiano Castelfranchi

ISTC-CNR & University of Siena, Italy
cristiano.castelfranchi@istc.cnr.it

Abstract. Observation is the basis of a very crucial form of Communication without words or special protocols. Efficient coordination – in humans but also in artificial Agents – exploits or should exploit not just 'observation' but more precisely this form of silent communication: when Agent X relies on the fact that Agent Y is observing her in order to let Y understand that p, i.e. for communicating to Y that p. The general theory of behavioral implicit communication (BIC) is presented. Its importance in social interaction is discussed. We will illustrate why this is crucial for mutual understanding and for comitment and norms keeping, imitation and learning, etc. *The message-sending paradigm* dominating CSCW, MAS, HCI, and H-Robot-I, is criticized. The relationship with the ill-defined but important notion of *stigmergy* (very used in ALife Agents) is analyzed.

1 What Behavioral Implicit Communication Is

Usually, communication is based on specialized signals linked to specific meanings. In order to use these signals, either we have to learn them or we refer to some sort of innate knowledge. The former is the case of spoken language, which we learn during our childhood, the latter is the case of expressive communication, in some way innate, so that we know what a smile means without previous learning. Above these two kinds of communication, there is a third one that we call *Behavioral Implicit Communication*, where there isn't any specialized signal, but the practical behavior itself is the message. BIC is very useful in a coordination context, where simply performing an action we send a message to our partner(s) in the interaction. This message could for example be intentional, i.e. the sender wants that the receiver knows that she is performing that action. However, this message presupposes a more primitive and basic substrate which is due to 'observation': the unilateral capability of the agent to observe the other's behavior and to "read" it, to understand what she is doing, what she intends and plans to do (her goals), or at least (in rule-based systems or neural net systems) to predict and expect her next position or action using this information for example (some sort of primitive "inference") for "anticipatory coordination"[1]. In other words, communication is based on and exploits 'signification' (the semiotic ability of cognitive agents; for example the ability to take 'smoke' as a sign of 'fire', or to ascribe thirst to a drinking agent) that goes beyond simple perception but it not necessarily used in and for communication.

Behavioral Implicit Communication is a wide-ranging phenomenon, which acquires more and more complexity as the communicative intention appears and become more

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 98 – 107, 2006.

important. In the weakest form of Behavioural Communication the agent is not acting in order to let the other understand what she is doing (this isn't one of her motivating intentions); she is simply aware of this possible result of her behaviour and lets it happen. This is a very weak form, because there is no real intention, the communicative result is just a known (not desired) side effect of the action. • The expected side-effects that I let happen could be either neutral or, at worse, bad but not so strong to make me renounce to the action.Before this form there is no communication at all but simply exposure and observation with interpretation activity by the observer ('signification').True BIC requires that the communicative effect should be meant on purpose by the agent, or it should be at least a side-intention of him. The agent anticipates that the action will have also a communicative meaning (additional result) and she likes this (it is a goal of her), but his action is not aimed to reach this communicative goal; not motivated by it. What is relevant here is that:

- The expected side-effects that I let happen is good but neither necessarily nor sufficient for my action.
- I know that you will see and understand and I also like/desire this.

This is a side or passive intention: the second step in behavioural communication.A third step occurs when I choose and perform the practical action also with the purpose to make you see and understand. The communicative effect is necessary (although not sufficient) for my action. This is full Behavioural Implicit Communication since communicating is part of my aims and motivates the action.The last step is when the behavior is performed only for communication and looses its practical purpose (or even the practical effect). In the last case, the act either is just faked (simulation, bluff) (which is very important in conflict coordination), or it is just a ritual, i.e. the action has fully become Non Verbal Communication or a conventional symbolic gesture with a practical origin. We have presented here the 'intentional' path in BIC evolution, when the communication is intentional. But there are more primitive forms of BIC for sub-cognitive Agents without true intentions, based on reinforcement learning or on selection (see Section 2). Also these forms are applicable to MAS: the first one is more suitable for BDI-like Agents, while the other two fit well with reactive or ALife Agents. We claim that in Human-Human, H-Agent, H-Robot [3], Ag-Ag,, Robot-Robot interaction [4] the possibility of communicating through an action is very interesting and could be the solution for several coordination problems, and, more than this, the interaction becomes natural and intuitive. We can briefly summarize the conditions for establishing BC via evolution, learning or reasoning:

> X modifies the environment (she leaves m/t) => Y perceives that this modification has a sort of "meaning" for him (we have signification and not yet communication) => X perceives that her action leads to a behavioural response by Y => X performs that specific action also in order to make Y perceive m/t.

This sequence, although very basic, can be used to take into account different kind of phenomena, with different degrees of complexity. This scheme describes what happens either at the intentional level (X intends that Y sees his traces and believes that…) or at the functional one, where X's behaviour has been selected by evolutionary function or reinforced by learning its effects on Y.In "functional" BIC the practi-

cal action - beyond the possible intention of the "source"- acquires (by evolution or design) a systemic function of informing the addressee. This is, for example, the case of Stigmergy [5], [6] in insects (and in robots or softbots) where communication is not an intention but just a function of the physical trace in the environment. Our first claim is that functional or reinforced BIC can be successfully generalized to the artificial Agents Interaction domain, where the robot or Agents can observe the traces left by the other agents in the physical or virtual environment, and can learn a correspondence between these and an appropriate action. Communication protocols, or natural language and expressive gestures and "faces" (NVC) will not be enough since they are not enough for human-human interaction and coordination!

2 The Stigmergic Over-Generalization

The notion of Stigmergy comes from biological studies on social insects, and more precisely the term has been introduced to characterize how termites (unintentionally) coordinate themselves in the reconstruction of their nest, without sending direct messages to each other. Stigmergy essentially is *the production of a certain behaviour in agents as a consequence of the effects produced in the local environment by previous behaviour* [6]. This characterization of Stigmergy is not able to discriminate between simple *signification* and true communication, and between prosocial and antisocial behavior. They would for example cover prey-predator coordination and a pilfer (unintentionally) leaving footprints very precious for the police.

In order to have "communication", it is not enough that an agent coordinates its behavior with the behavior or thanks to the traces of the behavior of another agent. Also this is an over-generalisation. Stigmergy is defined as "indirect communication through the environment". To us such a definition is rather weak and unprincipled. Any kind of communication in fact exploits some environmental 'channel' and some physical outcome of the act; *any communication is "through the environment"*! This cannot be the right distinction. The real difference is that in Stigmergic communication we do not have *specialized* communicative actions, *specialized* messages (that unambiguously would be "direct" messages because would be just messages); we just have *practical* behaviors (like nest building actions) and objects, that are *also* endowed with communicative functions. In this sense communication is not "direct" (special communicative acts or objects) and is "via the environment" (i.e. via actions aimed at a physical and practical transformation of the environment).

From our perspective, **stigmergy** is communication via long term *traces*, physical *practical* outcomes, *useful* environment modifications, not mere signals. Stigmergy is just a sub-case of BIC, where the addressee does not perceive the *behavior* (during its performance) but perceives other *post-hoc traces* and outcomes of it. In order a trace-based communication be stigmergy it is necessary that the perceived "object" be also a practical one and the originating action be also for practical purposes (like nest building).

Stigmergy is not only for insect, birds, or non-cognitive agents. There are very close examples also in human behavior. In animal stigmergy is non intentional, but intentional forms of it are possible. Consider a sergeant that – while crossing a mined ground –says to his soldiers: "walk on my prints!". From that very moment any print

is a mere consequence of a step, plus a stigmergic (descriptive "here I put my foot" and prescriptive "put your foot here!") message to the followers. Consider also the double function of guard-rails: on the one side they physically prevent car from invading the other lane and physically constrain their way, on the other side they in fact also communicate that "it is forbidden to go there" and also normatively prevent that behavior.

3 What Is Coordination and Why It Needs Observation and Signification

'Coordination' is that additional part or aspect of the activity of an Agent specifically devoted to deal and cope with dynamic environment interferences, either positive or negative, i.e. with opportunities and dangers/obstacles.
A 'common word' in fact means 'interference' (the action of an Agent Y could affect the goal of another Agent X)[1]; 'interference' can be either 'negative' (Y's action creates obstacles to X's action or damage her goals) or 'positive' (Y's actions realize X's goals or create opportunities for her actions). Thus X has to perceive (or infer) those 'interferences' in order to avoid or exploit them.

Coordination is not necessarily social (one can coordinate himself with a rolling stone); also when social, it is not necessarily mutual or cooperative as is usually assumed to be. The basic forms of coordination are: **Unilateral:** X just coordinates her own behavior with Y's or environmental dynamics, ignoring Y's coordination or non-coordination activity. **Bilateral:** X coordinates his behavior with Y's observed behavior; and Y does the same. *Bilateral but independent*: X coordinates his behavior with Y's observed behavior; and Y does the same in an independent way. **Reciprocal:** X coordinates his behavior with Y's behavior by taking into account the fact that Y is coordinating her behavior with X's behavior. **Mutual:** it is based on symmetric and interdependent intentions and mutual awareness (shared beliefs). Both X and Y wants the other to coordinate with his/her own behavior and understand that she intends to coordinate with her/his own behavior.

As we said, even when bilateral and reciprocal, Coordination is not necessarily cooperative. Also in conflict and war there is Coordination, and clearly is not cooperative and is not for cooperation. Prey and predator for example (consider a leopard following a gazelle) coordinate with each other: the leopard curves left and right and accelerates or decelerates on the basis of the *observed* path and moves of its escaping prey; but at the same time the gazelle jumps left or right and accelerates or not in order to avoid the leopard and on the basis of the *observed* moves of it. Their coordination moves mainly are 'reactive' (just in response to the previous move of the enemy), and for sure non-cooperative, non mutual. Moreover, they obviously are not communicating to each other their own moves (although they are very informative and meaningful for the other). This is an observation based but not a communication/message based (BIC) *reciprocal* coordination. **Observation for Coordination:** *One of the main functions of Observation in Agent living in a common word inhabited by other Agents is Coordination, while one of the main form of Coordination is Observation-based.* In fact, given the previous definition of 'Coordination', clearly in order to coordinated with a give event or act Ev X should perceive it or foresee it

thanks to some perceptual hints, 'index' or sign. In other word usually it is a intrinsic necessity of Coordination activity that of observing and interpreting the word in which X is acting pursuing its goals, and in particular observing Ev. In social coordination X must observe the other Agents' behaviors or traces for understanding what they are doing or intend to do.

In sum coordination is based on observation and - more precisely - 'signification'. **BIC for Coordination:** *A large part of Coordination activity (and social interaction) is not simply base on Observation but is BIC-based.* For example, clearly enough in **mutual coordination** not just Signification is needed but true BIC. Actually, since X wants that Y coordinates his behaviors observing and understanding what she is doing, she is performing her action also with the goal that Y reads it, i.e. she is communicating to Y - through her action - what she is doing or intends to do. But let's more systematically examine this.

In coordination the most important message conveyed by BIC is not the fact that I intend to do (and keep my personal or social commitments – which is crucial in cooperation), or my reasons and motives for acting, or the fact that I'm able and skilled, etc. (section 5.). It is more relevant communicating (informing) about when, how, where I'm doing my act/part in a shared environment, so that you can coordinate with my behavior while knowing time, location, shape, etc. In **unilateral Coordination:** Non-BIC-based Unilateral: Y coordinates (adapts) his own behavior to the interfering behavior of X, who does not perceive at all or does not care at all of those (reciprocal) interferences. In this case X's behavior is highly significant for Y (signification) but is not communication since X does not know or does not care of the fact that Y is observing her and interpreting her behavior. BIC-based Unilateral: only Y coordinates (adapts) his own behavior to the interfering behavior of X, but X knows and intends this, although she does not want to coordinate her own action with the other. X's behavior is communicative. In **bilateral Coordination:** Remember that this means symmetric-unilateral coordination; both Y and X coordinate their own action on the actions of the others but they ignore or do not intend that the other do the same. Again there can be no communication at all, but if one of the agent acts also in order the other perceives and understands what s/he is doing, there is BIC.In **mutual Coordination:** Both X and Y wants the other to coordinates with his/her own behavior and understand that s/he intends to coordinate with the her/his own behavior. As we said, mutual coordination, based on symmetric intentions and mutual awareness shared beliefs) entails and requires BIC: each coordination act (adaptation of the behavior) is a message to the other. Let us draw some conclusions on this point. Coordination is possible without any communication both in human and artificial societies ([1]; see also Franklin, http://www.msci.memphis.edu/~ franklin/coord. html). This is an important statement against common sense. However, usually coordination exploits communication. Since BIC is i) a very economic (parasitic), ii) a very spontaneous, iii) a very practice and rather effective form of communication just exploiting side effects of acts, traces, and the natural disposition of agents to observe and interpret the behavior of the interfering others, a rather important **prediction** follows.

One can expect that agents acting and perceiving in a common world will use a lot of BIC and will spontaneously develop it.

Actually a very large part of communication for coordination in situated and embodied agent exploits reciprocal perception of behavior or of its traces and products; i.e. it is just BIC. Even more, (second **prediction**):

Both in natural and in social systems a lot of specialized (conventional or evolutionary) signs derive from BIC behavior that has been ritualized.

This kind of observation-based, non-special-message-based communication should be much more exploited in CSCW and computer/net mediated interaction, in Multi-robot coordination, in Human-robot coordination, in MA systems. Behavioral communication *for coordination* (in particular Stigmergy) has some nice properties and advantages that deserve to be stressed. BIC is naturally and intrinsically 'situated' merged in concrete location, time, objects, thus it transmits this kind of information in a perceptual, immediate way without any special, abstract, arbitrary codification of this kind of information which will necessary be somewhat 'abstract'. While using BIC the addressee extracts the information directly from the environment (just 'observing' it) and it is 'ready to use' in that environment. This is also a possible memory charge advantage [8].

4 BIC and Some of Its Functions and Meanings in Human Agents

We are so used to BIC and it is such an implicit form of communication that we do not realize how ubiquitous it is in social life and how many different meanings it can convey. It is useful to give an idea of these uses and meanings, first of all just for understanding the phenomenon, second, because several of these uses can be exploited in HCI, in computer mediated H collaboration, in Ag-Ag interaction. BIC acts can convey quite different meanings and messages. Let's examine some on the most important of them for humans (also applicable to Agents).

i) "I'm able" or "I'm willing". The most frequent message sent by a normal behavior is very obvious (inferentially very simple, given an intentional stance in the addressee) but incredibly relevant:

> *(as you can see)* ***I'm able to*** *do, and/or* ***I'm willing to*** *do; since I actually did it (I'm doing it) and on purpose.*

There are several different uses of this crucial BIC message.

> Skills demonstration in learning, examines, and tests

When A is **teaching** something to B **via examples** and observes B's behavior or product to see whether B has learned or not, then B's performance is not only aimed at producing a given practical result but is (also or mainly) aimed at showing the acquired abilities to A. Also the behavior of the teacher is a BIC; its message is: "look, this is how you should do". Usually this is also joined with expressive faces and gestures (and with words) but this is not the message I'm focusing on.

> Showing, Exhibiting and Demonstrating

In general, if showing and exhibiting are intentional acts they are always communication acts.

> Warnings without words

This is a peculiar use of exhibition of power and dispositions that deserves special attention.

Mafia's "warning", monition. The act (say: burning, biting, destroying, killing) is a true act and the harm is a very true harm, but the real aim of this behavior (burning, killing, etc.) is communicative. It is aimed at intimidating, terrifying via a specific meaning or threat: "I can do this again; I could do this to you; I'm powerful and ready to act; I can even do worst than this". This meaning - the "promise" implicit in the practical act - is what really matter and what induces the addressee (that not necessarily is already the victim) to give up. The practical act is a *show down* of power and intentions; a "message" to be "understood".

ii) I did it"

Frequently, to **finish your food** is a message to guest: "I finished it, I liked it", as the guest wish and expects.

> The satisfaction of social commitments and obligations

Consider for example a psychiatric patient that *shows* to the nurse that he is drinking his drug as prescribed. (see later on social order).

BIC in Soccer

Team work is very much based on BIC. Consider a soccer game, a lot of actions and movements of a players are messages for some member of the same team, while are only meaningful sign for the opponents. For the opponents too there are BIC messages: feints and counter-feints, i.e. simulated actions. Also the ball and its trajectory or position can be a message for the others: *Stigmergic Communication* through the ball.

Silence as communication It is very well known that silence can be very 'eloquent'. In general, **doing nothing, abstaining from an action, is an action** (when is the result of a decision or of a reactive mechanism), thus it can be – as any behavior – aimed at communicating via BIC. The meanings of silence or passivity are innumerable, depending on the context and on the reasons for keep silence (or doing nothing) that the addressee can ascribe to "sender"; for example, indifference "I'm not involved, I do not care", "I do not rebel", – "I do not know", etc. …..or agreement. The most important social use, however, is for 'tacit agreements' that by definition are BIC-based. And tacit agreement or consent ('Qui tacet consentire videtur') is the way social conventions and informal norms emerge [10].

It is opportune that we spend some more word on BIC and Social Order.

BIC has a privileged role in social order, in establishing commitments, in negotiating rules, in monitoring correct behaviors, in enforcing laws, in letting spontaneously emerge conventions and rules of behaviors. If there is a 'Social Contract' at the basement of society this Social Contract has been established by BIC and is just tacitly signed and renewed. **iii) Imitation-BIC as convention establishment and memetic agreement**Imitation (i.e. repeating the observed beahavior of Y – the model) has several possible BIC valences (we already saw one of them).The condition is that Y (the model) can *observe* (be informed about) the imitative behavior of X. We can consider at least the following communicative goals:

a) In **learning-teaching via imitation**. X communicate to Y "I'm trying to do like you; check it: it is correct"
b) In **convention establishment and propagation**. "I use the same behavior as you, I ***accept*** (and spread) it as convention; I ***conform*** to it".

c) In imitation as emulation and **identification**: "I'm trying to do like, you I want to be and to behave like you" you are my model, my ideal".
d) In imitation as **membership**: "I'm trying to do like, you I want to be and to behave like you; since I'm one of you; I want to be accepted by you; I accept and conform to your uses (see –b)".

Let's focus a bit on the second BIC use of imitation, that is really important and probably ***the first form of memetic propagation through communication.***

X interprets the fact that Y repeats its innovation as a confirmation of its validity (good solution) and as an agreement about doing so. Then X will expect that Y will understand again its behavior next time, and that Y will use again and again it, at least in the same context and interaction. **iv) Fulfilling Social Commitments and Obeying Norms as BIC**This is another kind of demonstrative act, not basically aimed at showing power and abilities, or good disposition, but primarily intended to show that one have done the *expected* action. Thus the performance of the act is also aimed at informing that it has been performed! This is especially important when the expectation of X's act is based on *obligations* impinging on X, and Y is monitoring X's non-violation of his duty. Either X is respecting a prohibition, or executing an order, or keeping a promise. A second order meaning of the act can also be: "I'm a respectful guy; I'm obedient; I'm trustworthy", but this inferential meaning is reached trough the first meaning "I'm respecting, obeying, keeping promises". A **Social-Commitment** of X to Y of doing the act, in order to be really (socially) fulfilled, requires not only that agent X performs the promised action , but also that the agent Y knows this. Thus, when X is performing the act in order to keep his promise and fulfil his commitment to Y, he also intends that Y knows this.

(If there are no explicit and specific messages) ***any act of S-Commitment fulfilment is also an implicit communication act about that fulfilment.***

Notice that what is important for exchange relationships or for social conformity, is not that x really performed the act, but that y (or the group) believes so. **v) Fulfilling Social Commitments and Obeying Norms as BIC**One of the functions of norm obedience is the confirmation of the norm itself, of the normative authority of the group, and of conformity in general thus one of the functions of norm obeying behaviours is that of informing the others about norm obedience. At least at the functional level X's behaviour is *implicit behavioural communication*. Frequently, X is aware of this function and collaborates on this, thus he intends to inform the others about his respect of norms, or he is worrying about social monitoring and sanctions or seeking for social approval, and he *wants the others see and realise that he is obeying the norms*. In both cases, his conform behaviour is also an *intentional* behavioural/implicit communication to the others. Of course, X can also *simulate* his respect of the norms, while secretly violates them. At the collective level**, w**hen I respect a norm I pay some costs for the commons and immediately I move from my mental attitude of norm addressee (which recognized and acknowledge the norm and its authority, and decided to conform to it) while adopting the mental set of the norm issuer and controller [11]:

I want the others to respect the norm, pay their own costs and contribution to the commons.

While doing so I'm **reissuing** the norm, **prescribing** a behavior to the others and **checking** their behavior (expectation). Thus the meaning of my act is twofold:

"I obey, you have not to sanction me"; "Do as I do, norms must be respected".

This kind of routinary and tacit maintenance of social order is relevant also for MAS and HCI: doing what I promised or just passing the product of my activity to the other is a message, sending other explicit messages is not necessary and usually disturbing.

5 Concluding Remarks as for BIC and Agents

As we have shown, observation, and more specifically 'signification' (the capability to interpret and ascribe meaning to observed facts) is the basis of a very crucial form of Communication without words or special protocols. *Efficient coordination – in humans but also in artificial Agents – exploits or should exploit not just 'observation' but more precisely this form of silent communication*: when Agent X relies on the fact that Agent Y is observing her in order to let Y understand that p, i.e. for communicating to Y that p.BIC theory in fact looks quite relevant in several application domains in IT, especially with Agents, and for several important issues like:

- *The problem of social order and social control* in MAS, CSCW, virtual organizations. One cannot believe that social order will be created and maintained mainly by explicit and formal norms, supported by a centralized control, formal monitoring, reporting and surveillance protocols, etc. Social order will mainly be self-organizing, spontaneous and informal, with spontaneous and decentralized forms of control and of sanction [12] [13]. In this approach BIC will play a crucial role. Sanctions like the act of excluding, avoiding the bad guys will be messages; the same for the act of exiting. The act of monitoring the others behavior will be a message for social order; the act of fulfilling commitments, obeying to norms, etc. (as we already saw) will be all BIC acts. We should design those systems allowing for this perspective; for example allowing not only message sending but the possibility for reciprocal *observation* of behaviors or of their outcomes and traces. In this perspective the theory of 'observability' and the appropriate design of the 'environment' and of 'coordination artefacts' become crucial [8].- *Friendly and natural HM interaction* Machine collaboration and initiative (for example the so called 'Over-Help', see [14] should be based upon the possibility to "observe" and understand what we are attempting to do, and in anticipating or correcting it. The same holds in Human-Robot interaction [3] [4] [15]. It is not simply a matter of specialized and artificial messages (words or gestures); this seems rather unnatural. Also expressive NVC signals (faces, emotions – see [16]) are not enough. Even before this one should provide the robot – for example for coordination with humans in a physical environment – the ability to interpret human movements, understand them and react appropriately [17]. At that point the human action in presence of the robot will be performed also for its understanding, i.e. as a BIC message to it. Analogously the human should be in condition to monitor what the robot is doing and to intervene on it by adjusting its autonomy. At this point the robot behaviors – in front of the human – might become message for approval, help, coordination, and so on.

Acknowledgments

I would like to thank Frascesca Giardini for her contribution on BIC theory and for discussing several of these issues. I also thank Luca Tummolini with whom I'm working on 'Observation' and 'Coordination' theory, and Andrea Omici, A., Ricci, M., Viroli with whom we work on 'coordination artifacts' for Agents.

References

1. Castelfranchi C. "Modelling Social Action for AI Agents", *Artificial Intelligence*, 103, pp. 157-182, 1998.
2. Castelfranchi C. "Implicit Behavioral Communication in Imitation", *Second International Symposium on Imitation in Animals & Artifacts* at the *AISB'03 Convention*, 7th-11th April 2003 University of Wales, Aberystwyth, U.K. 2003
3. Giardini, F., Castelfranchi C. Will Ever Humans and Robots Coordinate Each Other Via Behavioral Communication? - EPSRC- Workshop, 2003.
4. Fong, T. Nourbakhsh, I. Dautenhahn, K. (2003), "A survey of socially interactive robots", In *Robotics and Autonomous Systems*, 42, pp. 143-166
5. Holland, H. Melhuish C. "Stigmergy, Self-Organization, and Sorting in Collective Robotics", *Artificial Life*, Vol. 5, N. 2, pp. 173-202, 1999.
6. Beckers, R. Holland, H. Deneuborg, J.L *"*From Local to Global tasks Stigmergy and Collective Robotics*I" Artificial Life IV,* Rodney A. Brooks and Pattie Maes (eds.), pag.181, 1996.
7. Buskens V. and Royakkers L. Commitments: A gametheoretic and logical perspective (2002). *Cognitive Science Quarterly*, in press.
8. Omicini, A. Ricci, A., Viroli, M., Tummolini L., "Coordination Artifacts: Environment-based Coordination for Intelligent Agents", *AAMAS 2004 Conference Proceedings*. 2004.
9. Castelfranchi C. & Falcone R., "To say and to do" virtual actions in the structure and recognition of discourse plans with regard to practical plans, *INTERACT '95*, Lillehammer, Norway, 27-29 June, 1995, pp.103-108
10. Castelfranchi C., Giardini, F. Lorini, E. Tummolini L."The Prescriptive Destiny of Predictive Attitude: From Expectations to Norms via Conventions", Proc. of *Cognitive Science Conference* 2003.
11. Conte, R. From conventions to prescriptions. Towards a unified view of norms. *AI&Law*, 1999, 7, 3, 323-40.
12. Conte R. and Castelfranchi C., *Cognitive and Social Action*. UCL Press, London, 1995.
13. Castelfranchi, C. Engineering Social Order. In A. Omicini, R. Tolksdorf, F. Zambonelli (Eds.) *Engineering Societies in the Agents World*. Springer, LNAI 1972, 2000, pp. 1-18.
14. Falcone R., The Human in the Loop of a Delegated Agent: The Theory of Adjustable Social Autonomy, *IEEE Transactions on Systems, Man, and Cybernetics*, Volume 31, Number 5, September 2001, pp. 406-418.
15. Kanda, T. Ishiguro, H. Ono, T. Imai, M. Nakatsu R. "Development and Evaluation of an Interactive Humanoid Robot "Robovie"". IEEE International Conference on Robotics and Automation (ICRA 2002), pp.1848-1855, 2002.
16. Breazeal C. "A Motivational System for Regulating Human-Robot Interaction". Proceedings of AAAI98, Madison, WI, 1998.
17. Nehaniv, C. Dautenhahn K. (eds), *Imitation in Animals and Artifacts*, MIT Press, Cambridge, Mass. 2000.

Simulating Working Environments Through the Use of Personality-Based Agents

Anne M.P. Canuto, André M.C. Campos , Araken M.Santos, Eliane C.M. Moura, Emanuel B. Santos, Rodrigo G. Soares, and Kaio A.A. Dantas

Informatics and Applied Mathematics Department,
Federal University of RN Natal, RN, Brazil, 59072-970
Tel.: +55-84-3215-3813
{anne, andre}@dimap.ufrn.br, {araken, eliane, emanuel, rodrigo}@ppgsc.ufrn.br, kaio@lcc.ufrn.br

Abstract. This paper presents a multi-agent simulation system, named SimOrg (Simulation of human organization), in which the personality aspect is incorporated in the internal functioning of the agents. The general architecture of the agents has been based on the theory of human personality from Theodore Millon [13]. The presented simulation has been used to simulate working environments, focusing on the training process within an organization. The main aim of this paper is to analyze how personality can affect the performance of the agent when simulating working environments.

1 Introduction

The development of agent-based systems has been largely increased in the last years [6],[8]. Examples of successful application of agent-based systems can be found in a wide range of applications, such as in electronic commerce, industry, web, among others. One of the applications of the agent technology is in simulation systems. Simulations are crucial in providing advice to natural resource managers, for training and management purposes, among others. The use of the multi-agent paradigm for building simulation leads to a more natural way to represents an observed system, where its dynamics is expressed in terms of interactions between autonomous entities (agents). Multi-agent systems are also able to model the spatial data (positions in the environment) of a system, which is hardly represented in some mathematical modeling approaches. Multi-agent simulations are becoming increasingly relevant in the simulation field [4], and can be applied to various areas [11].

Although the use of intelligent agents in simulating working environment seems to be very popular, there are some problems mainly in the natural representation of the human interactions. When simulating a working organization, for instance, it aims to investigate the individual behavior (micro level) of the agents in several possible situations as well as their interaction with other agent, resulting in emergent behaviors in the macro level. However, individuals performing the same role in a working organization can have very different performances, since they have different personalities and this affects the behaviors of these individuals. Most of the current applications of multi-agent simulation consider that the human behavior is based only on rational process.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 108 – 117, 2006.

Based on the aforementioned problem, this paper presents SimOrg, which is a multi-agent simulation, aiming to represent individual behaviors using some personality aspects [3]. In using personality-based agents, it is intended to perform a more natural simulation of a working environment, analyzing the influence of personality aspects in the performance of individuals and group dynamics.

2 Simulation of Human Organizations

Simulation systems have been widely used as a powerful tool to make some analyses of the decision making process within human organizations. In order to do that, it can be observed, through an artificial representation of the organizations, the behaviors of individuals which are part of this organization in a micro (individual) and a macro (social) level.

On the other hand, multi-agent systems have some important features, such as flexibility, collaboration and adaptation that are very useful in a wide range of applications [5],[4],[8]. Due to these features, several researchers have used the agent technology to simulate systems, leading to the multi-agent simulation field.

In the simulation of working environments, a multi-agent system can be used to represent an organization and agents represent an element of the problem. In other words, agents can be used to represent individuals within an organization [14].

The main aim of SimOrg is to define agent models that are able to represent personality and how they can interfere in the behavior of the agents and in the human interactions. There are some works aiming to represent personality's aspects of individuals in the agent architecture, such as in [5],[10],[12]. However, none of them use these aspects to simulate working environments. In this paper, the personality aspects are based on the theory of human personality and its working relationship from Theodore Millon [13]. This paper also tries to evaluate how personality can affect the relationship among different agents in a working environment.

3 SIMORG

SimOrg aims to develop a multi-agent simulation for human organizations in which each person will be represented by an agent [3]. Also, some aspects of the personality will be taken into account in the decision-making process of the agents. In order to take into account personality, some theories of the organizational psychology have been studied and it will be included in the architecture of the agent.

The organizational approach used to model the multi-agent system in the SimOrg (Simulation of Human Organizations) system is based on a hierarchical representation of structure and activities of an organization. This approach defines activities as composed of tasks, which aim to produce deliverables for being used as resource for the next activity. The simulation of this model aims to evaluate the execution of working projects through the integration of the work of several individuals (organization employees). It also takes into account formal and informal processes, including formation of groups, negotiation to the execution of tasks, manager activity as well as task

flow through the transference of responsibility at the different levels within the organization hierarchy.

4 Architecture of a SimOrg Agent

The unpredictable characteristics of individuals can be related to their personalities. Then, it is necessary to model the internal architecture of the agents in order to represent it. There have been several attempts to model personality within an agent, usually based on theoretical foundations about human personality, such as Big-Five [9],[12], Jung [1], among others.

In order to include some personality aspects in the architecture of the agent, the theory of Theodore Millon has been investigated and has been used as basis for the proposed architecture. The main reasons for choosing this theory to model personality within agents is that all aspects (topics and polarities) of personality can be easily modeled in the architecture of an agent. Also, as a result of a psychological test developed by Millon, it is possible to quantify some aspects of personalities (polarities of an individual) and this can be easily added to the decision making process of an agent. These features could not be found in other psychology theories.

4.1 Personality Theory of Millon

Millon proposed a measurement to express personality, which is based on the theoretical comprehension of the actions taken to reach the goals of an individual life [13]. Also, it takes into account the way to process information received from the environment. In this sense, it has been elaborated a tool to verify a dynamic configuration of interactions, representing three large areas: motivational aims, cognitive styles and inter-personal relations. Based on these areas, Millon stated that personality can be expressed through twelve bipolar attributes, distributed over these areas.

- The area Motivational aims includes three bipolarities, linking the ecological and evolutionist theories, which are: openness versus preservation, modification versus accommodation and individualism versus protection [2].
- The cognitive styles can be found in the evolutionist perspective as well as in contribution of authors such as Jung and Myers. The main aim is to evaluate the way an individual can process information. The bipolarities can be defined as extroversion versus introversion, feeling versus thinking, reflection versus affectivity and systematization versus innovation.
- The inter-personal relationship: it evaluates the style of the relationship of an individual with the others. The bipolarities are: shyness versus communicability, doubt versus security, discrepancy versus conformity, control versus submission and satisfaction versus dissatisfaction.

At a first sight, Millon and Big five theories can look equivalent. However, Millon can be seen as an extension of the Big Five theory [2]. Although the Millon theory incorporates ideas from the Big five model (attributes like introversion/extroversion), they differ in the conceptual level. While Big five can be seen as a pragmatic model (personality classification is based only on observed behavior), Millon also incorporates ideas coming from evolutionist and biological theories of personality.

4.2 The Proposed Architecture

As already mentioned, each agent in the SimOrg simulation represents an individual within an organization. The first step towards the representation of personality within the internal structure of the agent is to define the architecture of this agent. The architecture for the SimOrg agent is shown in Fig. 1. As it can be observed, the defined architecture contains not only modules commonly used in a common agent architecture, such as perception, planning, goal definition and action, but also it is composed of a base, defining the artificial psychological profile of the agent.

As personality can be seen as the way in which an individual interacts with the environment, it is represented as an information base, or personality base. It is composed of the three aforementioned topics of the human personality (motivational aims, cognitive modes and inter-personal relations). This personality base has strong influence in the definition of the goals of an agent (goal definition module) as well as in the choice of the actions to perform the chosen goal (planning module).

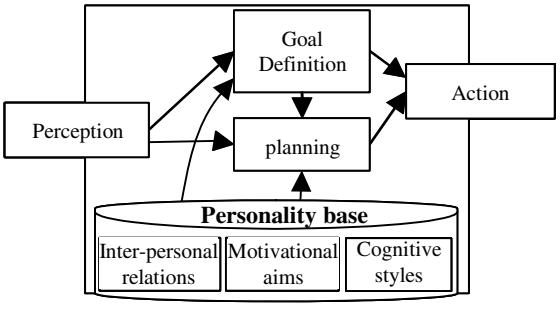


Fig. 1. The general architecture of a SimOrg agent

In considering the personality base as a common knowledge base, this agent can be seen as a common BDI (Beliefs, desires and intentions), in which based on its beliefs, an agent can choose its goals as well as the needed actions to reach this goal. The use of personality will help to transform the modeling in a more natural one, in which the personality of an individual should and will be taken into account.

The way in which personality can influence in the decision making process of an agent is through the values associated to all bipolarities included in the three aforementioned areas. All bipolarities can be seen as attributes, considering both features in a complementary way (for instance, 90% of extroversion and 10% of introversion). These values will be used to define the probability of activation of rules (goals or actions) for an agent. It is important to emphasize that all polarities will be taken into account in this definition. These probabilities can be seen as weights that are taken into account in the decision making process. For instance, in order to reach a goal, an agent can have three possible set of actions to perform, the one which has the highest probability (weight) will be chosen. In this sense, it means that the lowest the probability is, the more distant the rule is from the personality of an agent. However, even rules with low probabilities can be chosen. This allows representing unexpected actions of an individual, which is a characteristic fact of the human beings.

5 The SimOrg Scenario

In order to analyze the influence of personality in the performance of the SimOrg agents, it has been defined a scenario for a working organization, where agents could not only interact and act according to their own personality, but also follow a working process (organizational rules). The scenario was based on the experience of the authors in developing software. In this sense, the fictitious organization is a software development company. In this scenario, the chosen development process is an iterative one based on agile methodologies, which was simplified for testing purposes. Agile methodologies are ideal for testing the proposed architecture since individuals and their informal communication are very important for them.

The scenario was then represented as having formal, informal interactions. The task actions are: design, estimate time, code, test code and validate design. Those actions were organized into four activities (time estimation, design, development and project evaluation), each one having an individual or group responsible for. The communication acts follows a small subset of an ACL structure, composed of performative, sender, receiver and contents.

Although this is enough for representing communication between agents, it was observed that human communication is not necessarily binary (to send or not to send a message). In order to model working human organizations, it is also necessary to represent how good an individual performs an action and the importance that an individual gives for those actions. As a consequence, two other components were introduced in our ACL, which are: quality and importance, whose values are functions of the individual personality attributes. In this way, an agent is also able to react to a received message according to its quality and importance.

6 SimOrg Simulation

The implementation of the SimOrg simulation, based on the MAVIS framework [4], is divided into two main modules, which are: the collaborative web-based environment and the simulation. The simulation module was developed using Java programming language as well as XML (eXtensible Markup Language) and RMI (Remote Method Invocation) technologies.

Scenarios are described using XML files that are responsible for creating simulation models. These files are divided into two main parts, which are: organization and process. In the organization part, the structure of the organization is described. In this part, all the distributions of the agents in the groups as well as the association of roles to the agents within the organization are described. On the other hand, the process part describes how the actions of the agents are organized into activities as well as the responsibilities of each group and which agents are involved in the execution of the activities.

At this stage of the implementation, a stable version of the Simulator is ready and on use. Some tests have already been done in the current version and it is now being validated by possible users. Also, an interactive graphic interface has been built in order to fulfill the requirements of an interactive simulation, as the SimOrg simulation is. The main graphic elements of the simulator are the following: the different data

visions of the fictitious organization and the actions that the users could use to interact with the simulations.

6.1 The Dynamic Aspects of the Agents

As already mentioned, in the SimOrg simulation, agents have a personality base, which was based on the theory of personality of Millon. The theory was used to map the suitableness of the role of an agent within the organization to its personality characteristics. Indeed, personality plays an important role in the tasks performed by an individual [7]. In order to evaluate the performance of the agents, a parameter, called quality of the actions, is defined. In the simulation model, time for accomplishing a task is directly proportional to the quality of the actions performed during the task execution. In other words, actions with low-level quality will result in late tasks. The quality of an action is defined based on two main factors, which are.

- Personality suitability of an activity: it relates the personality parameter with the activity of the agents. In this sense, optimal profiles were defined for each activity. During the execution of an activity, the personality of the agent is compared with the most suitable profile for that activity. The result represents the suitability of the agent to perform the current activity.
- Motivation parameter: it represents the motivation level of the agents. The motivation level is a function in which factors like personality, fatigue and work pressure in the working organization have a negative influence. Fatigue expresses the amount of time that an agent would be working. The more fatigue an agent has, the less its motivation level is. On the other hand, work pressure is related to the speed (fast or slow) of the activities of an agent. According to experts in the human resources field [1], it works as a modifier of the motivation level. In the beginning, the more pressure an agent has, the more motivated it is. However, during the execution of the activities, there is a steeper decrease in the motivation of the agents.

In order to evaluate the performance of the agents, the optimal execution time was estimated for all activities of the organization. An optimal execution time is reached by the agents with the best quality and highest motivation level. Also, it is possible to evaluate the individual and group performance of the activities, since the optimal execution time is provided to individuals and to activities.

6.1.1 The Use of Stereotypes

Through an analysis of the theory of personality, some personality profiles (or stereotypes) within a working organization were identified. These stereotypes represent standard profiles of individuals within an organization. The main stereotypes which were defined by the experts (psychologists) are leader or submissive, optimistic or pessimistic, cooperative or individualist, objective or subjective, among others (see [1] for the complete list of defined stereotypes). In addition to that, an individual can have more than one stereotype, being, for instance, leader and optimistic.

According to the experts, a mapping from the stereotype to the twelve bipolarities which are part of the personality module of the agents is possible. This mapping has been done based on a form, elaborated by the experts (psychologists), which were

filled by several individuals working in several software development companies (software developers, architects, coder, etc.). After that, the experts transformed them into weights to the bipolarities. Also, for individuals containing more than one stereotype, the mapping was done taking into account the influence of each stereotype in the behavior of the agents [1].

The main aim of using stereotypes was to make the process of choosing the personality of the agents automatically. Otherwise, users would have to type the weights of all bipolarities for all agents of the company. In using stereotypes, the users only choose the main features of the agents and the transforming process is made automatically.

In order to implement the choice of the stereotypes of the agents, several questions, elaborated by the experts, are asked to the user. These questions do not ask the direct subject. For instance, there is no question like "are you leader or submissive?". However, the question is "In a working group, would you prefer to make the decisions or to let the others do it?". The idea is not to bias the user to always answer, for instance, leader, which can be considered as a good feature for an agent, instead of submissive.

6.1.2 User Actions

There are several ways that a user can interact with the Simulator, either changing the organization structure or changing the motivation level of the agents. These interactions can be done through the use of user action. The user actions which can change the organization structure are Employ (employ a new individual), Dismiss (discharge an agent) or Change roles (change the activities of the agents). On the other hand, the user actions which can change the motivation level of the agents are: raise salary, decrease the work speed, increase the work speed, promote and individual recognition. It is important to emphasize that both types of actions affect the motivation level of the agents. However, the first ones also change the organization structure.

7 An Analysis of the Simulation Results

As already mentioned, the main aim of this simulation is to obtain results of the activities of the agents in order to evaluate the performance of each agent. Also, it aims to analyze the influence of personality in the performance of the agents and groups. At the end of the simulation, it is possible to investigate the influence (personality) of each individual in relation to its activities.

It is important to emphasize that, based on the analysis of the individuals (agents), it is possible to analyze the performance of groups of agents. As already mentioned, an activity can be performed by several agents. In this simulation, it is possible to analyze the performance of an activity (group of agents) and of individuals (agents). In this sense, it is possible to detect which agent is having the most negative influence in the performance of the activity. In having this information, users can do some actions in order to improve the performance of the individual, improving, as a consequence, the performance of the activity.

In this analysis, the performance of different agents is investigated, having different stereotypes, in executing the same activities. Also, no user action is allowed since it is aimed to analyze the influence of personality in the execution of the activities.

7.1 Individual Performances

In this part of the analysis, the individual performances will be considered. Fig. 2 represents the performance of four agents with different stereotypes, which are individualist, optimistic cooperative and perfectionist. The x axis represents the number of epochs (units of time, which can be hours, days or weeks, depending on the working organization), while the y axis represents the quality of the actions performed by the agents. All the agents are executing the development activity (design). The idea is to investigate the influence of personality in the execution of the same activity.

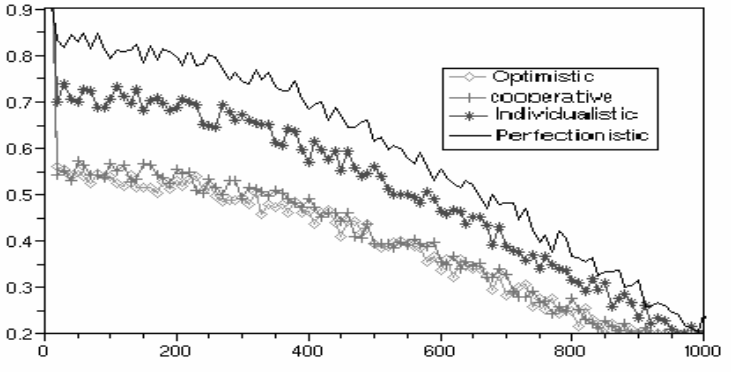


Fig. 2. Performance of the agents when performing the manager role and the design activity

As it can be observed from Fig. 2, the perfectionist agent had the highest quality of action, followed by individualist, cooperative and optimistic. In epoch 400, for instance, the quality of the perfectionist agent (0,75) is almost double of the optimistic agent (0,445). This means that, at this unit of time, the execution time of an optimistic agent is almost twice the execution time of the optimistic agent. As time passes, this difference tends to decrease.

Another important fact is that as the number of epochs (time) increases, the quality of the actions of all agents decreases. This is because when no stimuli are given to individuals, their level of motivation tends to decrease. The main idea of using user actions is to improve the quality of the actions through stimuli to the agents. It is important to emphasize that users actions should have different influences in the motivation level of the agents, according to its personality. However, in the current version of the simulation, user actions have the same influence in the motivation level of all agents, even when they have different personalities. These differences have been analyzed by experts and will be soon inserted in the simulation.

7.2 Group Performance

In the second part of this analysis, it is aimed to evaluate the performance of groups of agents having different personalities executing the same activity. In order to do that, the design activity was also considered. It was defined in the simulation that three agents are needed to execute this activity. Then, based on the results obtained in the

last subsection, for the first case, the three best personalities (perfectionist, individualist and perfectionist) were chosen to compose the designer group. In the second case, the three worst personalities (optimistic, cooperative and optimistic) were chosen to compose the designer group.

Fig. 3. Execution time of two teams of agents executing the design activity

Fig. 3 illustrates the real and optimal execution times for both cases. It can be observed that both teams reached the estimated without completing the activity. However, the team composed of the best three agents has an execution time much closer to the optimal execution time than the other team. If, for instance, a user decides to change the individualist agent of the best agent team for another perfectionist, the execution time would be even closer of the optimal execution time.

It is important to emphasize that even for the best personality analyzed, the quality of the action was not 1. This is because the quality of an action is based on the personality suitability and motivation level.

8 Final Remarks

This paper presented SimOrg, a simulation tool for working environments which have used personality-based agents to represent individuals in the organizations. It could be observed that the personality of an agent can have influence in the quality of the actions and, as a consequence, in the execution time of the activities. Also, groups composed of personalities which are not suitable for the task do not have a good performance, spending more time to execute the activities than the other groups. In a working environment, execution time is very important to define the performance of individuals and groups. In this sense, it can be concluded that personality was very important to the performance of the simulated organization. Furthermore, it is very important to include this parameter in the functioning of the agents, which has been done by the SimOrg system.

In this paper, the influence of the personality in the execution time was analyzed. However, in working organizations, there is a wide range of analysis that can be done. For instance, an analysis the influence of one personality in the others could be very useful. In this sense, the execution of an activity would not only be defined in terms of the personality, but also to the affinity of the personalities executing the activity (inter-personal dynamics). This is the subject of an on-going research.

References

1. Alchieri, J. C, Campos, A and Canuto, A. The effects of personality in the performance of individuals of a working organization. Tech Report, UFRN, 2005.
2. Alchieri, J. C. Models of Millon personality styles: An adaptation of Millon personality styles . PhD Thesis. Federal University of Rio Grande do Sul, Porto Alegre, 2004
3. Alchieri, J. C, Campos, A and Canuto. SimOrg: Simulation of Human Organizations. In: "Psichological Evaluation of Organizations", ed. Vetor. In press, 2006.
4. Campos, A. and Hill, D., "An Agent-Based Framework for Visual-Interactive Ecosystem Simulations". In Transaction of SCS, 15(4):139-152, December, 1998.
5. Campos, A., Lustosa Neto, V. P., Costa, A., and Gonçalves, L.M., "Introducing Personality in Game Characters", Proceeding of the 2nd Brazilian Workshop on Games and Digital Entertainment, 2003.
6. Campos, A., Santos, E., Canuto, A., Soares, R. and Alchieri, J. A Flexible Framework for Representing Personality in Agents. AAMAS 2006, v.1, pp. 97-104.
7. Church, A., and Waclawski, J. The relationship between individual personality orientation and executive leadership behaviour. Journal of Occupational and Organizational Psychology. 71(2):99-126. June 1998.
8. Deravi, F., Fairhurst, M., Guest, R., Canuto, A., Mavity, N., "Intelligent Agents for the Management of Complexity in Multimodal Biometrics". In Journal of Universal Access in the Information Society. Springer-Verlag. 2(4):293-304, 2003.
9. Digman, J. M., "Personality Structure: Emergence of the Five-Factor Model", in Annual Review of Psychology, 41:417-440, 1990.
10. Egges, A., Kshirsagar, S., and Magnenat-Thalmann, N. 2004. Generic personality and emotion simulation for conversational agents. J. Comput. Anim. Virtual Worlds 15, 1--13.
11. Gilbert, N. e Troitzsch, K.G., Simulation for the Social Scientist. Open University Press. 1999.
12. Howard, P.J., Howard, J.M. (2001). The BIG FIVE Quickstart: An Introduction to the Five-factor Model of Personality for Human Resource Professionals, Center for Applied Cognitive Studies, Charlotte, North Carolina, www.centacs.com/quickstart.htm.
13. Millon, T (1997). "The MIPS: Gauging the Dimensions of Normality. In The Millon inventories: clinical and personality assessment. (p. 498-522). New York. Guilford.
14. Schmidt, B. The Modelling of Human Behavior. SCS Publications. Erlangen, Alemanha. 2000.

GAPatrol: An Evolutionary Multiagent Approach for the Automatic Definition of Hotspots and Patrol Routes

Danilo Reis, Adriano Melo, André L.V. Coelho, and Vasco Furtado

Graduate Program in Applied Informatics (MIA), University of Fortaleza (Unifor)
danilo.reis@fujitec.com.br, aanmelo@hotmail.com,
{acoelho, vasco}@unifor.br

Abstract. In this work, we present a novel evolutionary multiagent-based simulation tool, named as GAPatrol. Such system is devoted to the specification of effective police patrol route strategies for coping with criminal activities happening in a given artificial urban environment, which, in turn, mimics a real demographic region of interest. The approach underlying GAPatrol allows for the automatic uncovering of hotspots and routes of surveillance, which, in real life, are usually discovered by hand with the help of statistical and/or specialized mapping techniques. The qualitative/quantitative results achieved by GAPatrol in two scenarios of study are discussed here, evidencing the potentialities of the novel approach as a promising decision-support tool for police patrolling.

1 Introduction

Police patrolling is an important instrument for implementing preventive strategies towards the combat of criminal activities in urban centers, mainly those involving violent practices. One hypothesis underlying such preventive work is that, by knowing where the occurrences of crime are currently happening and the reasons associated with such, it is possible to make a more optimized distribution of the (human) police resources available to control (and, ideally, to decrease) the overall crime rates. One big problem in this context, however, is that understanding criminal mapping activities, even using geographic information systems, is a non-trivial task. In addition, real-life experiments in this domain cannot be performed without high risks, as they may result in loss of human lives. As a consequence, the deployment of simulation systems for decision support in the public-safety domain comes to be very pertinent.

With this in mind, some prominent studies have been conducted exploiting the theoretical/empirical resources made available by the area of multiagent-based simulation systems [3][6][9]. One such research work was developed by Machado et al. [4], and further complemented by Almeida et al. [1], within the theme of multiagent patrolling (MAP), having as basic motivation to stipulate generic rules for the construction of adequate multiagent-system (MAS) architectures as well as the criteria to evaluate them. Following another direction, Winoto [10] exploited the multiagent paradigm for characterizing some important crime features. In his work, an economic perspective upon crime was elaborated and the notion of impunity, which seems to be a crucial factor for the increase/decrease of crime rates, was analyzed from the crime repression viewpoint. The preventive aspect, however, was overlooked by the author.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 118 – 127, 2006.

One crucial aspect related to the MAP task that deserves to be more systematically investigated may be described as follows: How to design better police patrol routes for crime regulation, taking into account the peculiarities of the patrolling scenario under consideration as well as the constraints usually imposed on the amount of human resources available to perform preventive monitoring. Associated with such an "optimal design of patrol routes" (ODPR) problem are two important, correlated questions, namely (i) How far from these optimal patrolling routing strategies are those actually adopted by human police managers?; and (ii) How complex do such optimal patrolling routes need to be in terms of their total lengths and urban area coverage?

In this work, we take a step in the direction of tackling the ODPR problem and its associated issues by presenting a novel evolutionary multiagent-based simulation tool, named as GAPatrol. This tool provides an automatic means for the delimitation of important hotspots of surveillance (i.e., places where crime occurrences are more frequent [5][7]) as well as the specification of effective routing strategies (i.e., time and space disposition of the human police resources) for preventive patrolling over an artificial environment that mimics a real demographic region of interest. The conceptualization of GAPatrol was directly inspired by the increasing trend of hybridizing MAS with evolutionary algorithms (EAs) in such a way as to combine their complementary properties [8]. Therefore, the evolutionary multiagent model behind GAPatrol can be regarded as an alternative vehicle for uncovering hotspots and patrol routes, which in real life are usually discovered by hand (i.e., under the supervision of human experts on the public-safety domain), sometimes with the help of statistical and iterative mapping tools. The qualitative/quantitative results achieved while carrying out some GAPatrol simulation experiments are discussed here, considering two distinct crime scenarios over the same urban territory. Such results testify the potentialities of the novel approach as a promising decision-support tool for police patrolling.

The article proceeds as follows. Initially, we characterize the entities comprising our simulated urban society and describe configuration issues related to the adopted multiagent simulation environment. Subsequently, we provide details about the evolutionary engine behind GAPatrol and, afterwards, we present the simulation scenarios and discuss the quantitative/qualitative results achieved with the GAPatrol experiments. The last section concludes the paper and brings some remarks on future work.

2 GAPatrol: Multiagent Society and Configuration Aspects

The entities that take part in our simulated multiagent society are described as follows. There is a set of N_p police teams available, each one associated with a monitoring route passing through some special locations of the urban territory considered. There is no distinction, in terms of skills, between the police officers allotted to the different police teams. We also assume that the teams patrol intermittently and the speed of their patrol cars are the same, meaning that the time spent by a given team in a given location will depend solely on the size of its route—different teams may be associated with differently-sized routes which in turn can overlap and/or share common points of surveillance. The special locations to be patrolled are referred to as

targets, which can be differentiated with respect to the type of commercial/entertainment establishment they represent (such as drugstores, banks, gas stations, lottery houses, squares, and malls). In all, there are N_t targets in the simulated urban territory.

Furthermore, there is a set of N_c criminals representing the actors that frequently try to commit the crimes. Each criminal is endowed with a limited sight of the environment, measured in terms of grid cells. For instance, with a vision of 1,000 meters, if each cell has 100-meter sides, the radius of the criminal's sight will be 10 square cells around him/herself. We assume that there is no chance of having a criminal being arrested and jailed, which implies that the number of criminals is always constant. Contrary to the police officers, who are homogeneous in their skills, each criminal offender has a personality, which, in turn, determines the types of places he/she more frequently selects as targets for committing crimes. Moreover, the personality can vary over time, passing from novice to intermediate to dangerous, according to the success of the criminal in committing crimes. For each pair (type of personality, type of target), there is a certain probability for committing a crime, as shown in Table 1. The underlying logic is that a dangerous criminal has a higher probability to seek out banks than a novice, aiming at obtaining higher returns in money and due to his/her higher level of expertise.

Having probabilistically selected the next type of target, we assume that the criminal has the knowledge necessary to localize the closest exemplar target on the map, moving straight towards such point. The time expended to reach the target is calculated based on the speed of the criminal and the distance to the target. The shortest period of motion, considering all N_c criminals, is taken as a reference, so that the criminals are allowed to move only during this time period. Finally, the decision to whether or not commit a crime is made based on the existence of one or more police teams within the radius of the criminal's sight. If the offender decides to not commit a crime, then he/she will select a new target to approach, leaving the current location. Otherwise, we assume that a crime will be committed.

Table 1. Probability of approaching a target for different types of criminal personalities

	Square	**Drugstore**	**Lottery House**	**Gas Station**	**Mall**	**Bank**
Dangerous	10%	15%	15%	10%	15%	35%
Intermediate	20%	30%	30%	20%	-	-
Novice	50%	30%	20%	-	-	-

The criminal behavior as shown above was designed to be intentionally very simple in such a manner as to facilitate the *a posteriori* analysis of the patterns that emerge from the simulation. A more complex criminal behavior could be simulated, endowing the criminals with the capacity of learning to identify promising targets (hotspots) in line with their experience.

Our multiagent simulation tool runs atop the Repast environment developed at the University of Chicago [3]. Each set of routes designed by a GAPatrol chromosome (see below) gives rise to a set of N_s simulation executions in order to evaluate the

crime prevention performance of such set of routes. Each simulation instance runs 3,000 ticks, which would roughly correspond to one month of real-life events.

3 GAPatrol: Evolutionary Engine

Genetic Algorithms (GAs) [2] are general-purpose search and optimization algorithms that comply with the Darwinian natural selection principle and with some principles of population genetics to efficiently design (quasi-)optimal solutions to complicated computational and engineering problems. Such metaheuristics maintain a population of chromosomes, which represent plausible solutions to the target problem and evolve over time through a process of competition and controlled variation. The more adapted an individual is to its environment (i.e., the solution is to the problem), the more likely will such individual be exploited for generating novel individuals. In order to distinguish between adapted and non-adapted individuals, a score function (known as fitness function) should be properly specified beforehand in a manner as to reflect the main restrictions imposed by the problem.

In the case discussed in this paper, a customized GA model is employed to cope with the ODPR and its associated issues, making use of the multiagent simulation model described previously. Indeed, this is a very demanding problem to be investigated through an evolutionary approach, due to the dynamism implied by the criminal agents' non-deterministic behavior. In what follows, we formalize the ODPR problem and describe the main components of the GAPatrol evolutionary engine.

Let $T = \{T_j\}, j = 1,\ldots,N_t$ be the set of targets considered in the urban environment, each one related to a pair of physical coordinates (x_j, y_j), $P = \{P_i\}, i = 1,\ldots,N_p$ be the set of police patrol teams available, and $R = \{R_i\}, i = 1,\ldots,N_p$ be the set of patrol routes, where each route R_i should be interpreted as a sequence of one or more target coordinates $\{(x_k, y_k)\}$, $k \in \{1,\ldots,N_t\}$. Let N_{cr} denote the number of crime occurrences in a given period of observation, given as a function of $T, P,$ and R, then the ODPR problem can be formalized as follows: $R^* = \arg\min_R N_{cr}(T,P,R)$. That is, to find the set of patrol routes R^* so as to minimize the number of crime occurrences, taking into account the nuances of the patrolled scenario and the profiles of the criminal agents.

In order to better discriminate between the GA individuals' capabilities and, thus, to prevent the premature convergence problem, we have resorted to a fitness function that is based on the fitness scaling mapping operator [2]. For a given set of patrol routes codified as a GA individual, say R^+, its fitness value is calculated as:

$$fit(R^+) = 100 \times \frac{\left(\mathrm{MAX}_{cr}^{g} - N_{cr}(R^+)\right)}{\left(\mathrm{MAX}_{cr}^{g} - \mathrm{MIN}_{cr}^{g}\right)},$$

where MAX_{cr}^{g} (MIN_{cr}^{g}) denotes the highest (lowest) number of crime occurrences among the individuals of the *g*-th GA generation.

Considering the codification of the solutions into the chromosomes, as mentioned before, each GA individual represents a given set of routes, whose lengths are defined adaptively and may vary from one to N_t. By this means, no *a priori* restriction is imposed on the length of the route R_i of each police team and the targets it may include. As depicted in Fig. 1, each chromosome should be interpreted as a sequence of (P_i, T_j) pairs, in which each police team and target is referenced by its corresponding index number (i or j). Such index numbers are codified via the binary alphabet. Moreover, in order to promote parsimony in the definition of the lengths of the routes, there are some reserved index values for the police teams and targets that could be exploited by a certain chromosome string to signalize the pruning of the number of resources effectively used. Such genetic material, although present as part of the given chromosome, will not be "expressed" in the decodification phase, giving rise to smaller sets of routes. In this sense, there are no individuals considered as invalid (unfeasible) solutions for the ODPR problem.

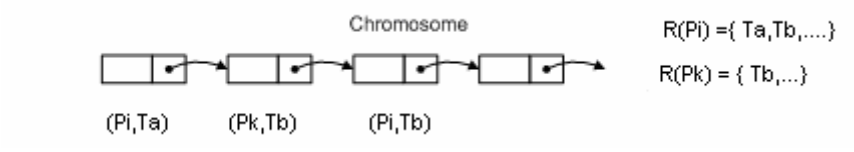


Fig. 1. Interpretation of a GAPatrol chromosome

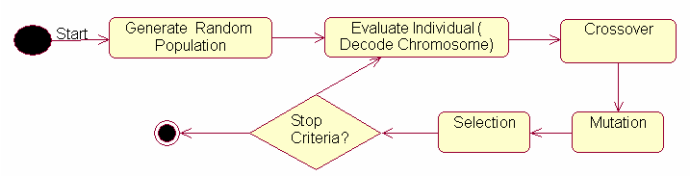


Fig. 2. Flowchart of the GAPatrol evolutionary engine

Figure 2 presents the flowchart of the GAPatrol evolutionary engine. The initial population of route sets is randomly created according to a uniform distribution over the values of all genes. The fitness value of each chromosome (set of routes) will be calculated taking as basis the average number of crimes achieved in N_s executions of the multiagent simulator. After that, according to the roulette wheel selection operator [2], some individuals are recruited for mating (via a crossover operator) and producing offspring. The latter may be target for the simple mutation operator. Finally, the current best chromosomes (from both parents and offspring) are selected deterministically for comprising the next generation of GA solutions. This process is repeated until a stopping criterion is satisfied, namely that a certain number of generations is reached. Then, the best individuals (sets of patrol routes) are presented to the user as the (quasi-) optimal solutions to the considered ODPR problem instance.

4 Experimental Methodology and Results

In order to evaluate the performance of the GAPatrol approach while tackling the ODPR problem and its associated issues, some simulation experiments were carried out having as basis two distinct scenarios defined over a simulated urban environment that mimics a well-known neighborhood of Fortaleza, a metropolis at northeast of Brazil. The simulations were conducted over a network of 10 desktop computers (Pentium IV, 2.8 GHz, 512 Mbytes RAM), where each GA execution spent about 3½ hours. In such study, the multiagent simulation parameters defined in Section 0 were set as follows: $N_s = 10$, $N_t = 41$, $N_p = 6$, and $N_c = 15$. Likewise, the GA meta-parameters were set arbitrarily as follows [2]: $\lambda_c = 95\%$ (crossover rate), $\lambda_m = 5\%$ (mutation rate), $\lambda_{pop} = 30$ (population size), and $\lambda_g = 100$ (maximum number of iterations). Since the criminals begin the simulations as non-experts (i.e., novices), the points that should be targeted more frequently by them are drugstores, squares, and lotteries. Figure 3 presents the physical disposition of all 41 targets over the environment as well as the points of departure of the criminals in the two scenarios.

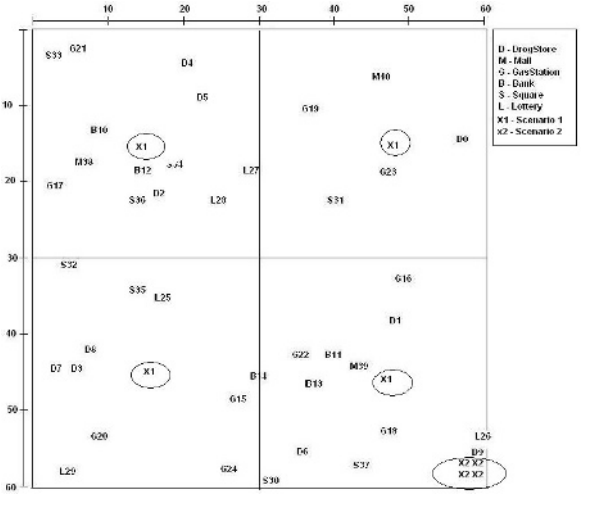


Fig. 3. Physical location of the targets (in terms of grid squares) and points of departure of the criminals in both scenarios

The first scenario was devised as a controlled scenario where the points of departure of the criminals were localized strategically in the middle of the four quadrants of the environment. In this way, the criminals tend to focus on those targets located in the vicinity of their points of departure, as these points already satisfy their novice expectations. In the future, this leads to a concentration of the hotspots, whose numbers also tend to be small. Such scenario is regarded as "controlled" inasmuch as it simplifies the process of routes generation by the expert, since he/she can more easily recognize the hotspots and then create an effective patrol strategy. Conversely, in the

second scenario, the criminals start out from a unique source, localized in the lower right quadrant. This configuration forces the criminals to initially roam around the area, leading to a more dispersed distribution of the hotspots. By now, the definition of an effective patrol route is not as easy as before.

In what follows, we present the results achieved with GAPatrol in both scenarios and contrast its performance with that obtained by an expert manager assisted only with the MAS simulator. Finally, a more qualitative discussion is conveyed.

4.1 Empirical Results

Figure 4 depicts the best sets of routes designed by GAPatrol and by a police expert manager considering both scenarios of study, while Table 2 presents the configurations, in terms of sequences of target indices, of the GAPatrol best sets of routes as well as their associated numbers of committed crimes. At the end of this subsection, we provide a more detailed contrast between the performances exhibited by both approaches; for now, we focus on some nuances related exclusively to GAPatrol.

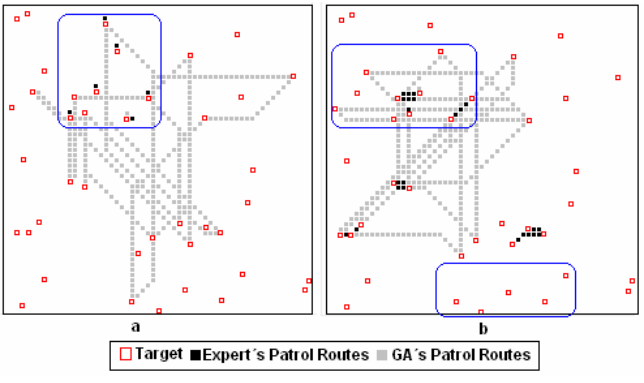


Fig. 4. Best police sets of routes for the first a) and second b) scenarios. Circles with continuous borderlines indicate the targets with more crime occurrences.

Table 2. Configuration, in terms of target indices, of the GAPatrol's best set of routes

Solution	Team 0	Team 1	Team 2	Team 3	Team 4	Team 5	# of Crimes
Scenario 1	34	36	19,27,22,4,14	24,14,2,39,4,0,31, 24,5,2,28,26,28	28	25,2,35,12,27,1 3,38,36,31,17	135.0
Scenario 2	35,28,8,31	2,36	5,15,3,25,19,14 ,25,12,22,8,25	28,19,7,35,25	34	25,31,10,27,31, 17,31,12,21,14	445.0

First, as one can notice, the sets of routes designed by GAPatrol in both scenarios tend to concentrate on subareas with a high density of targets. As well, such patrol routes did not need to encompass all the targets in the territory in order to be collectively effective. These two results are also confirmed by observing three important statistics conveyed in Table 3: 1) "distance", i.e. the sum of the lengths of the designed routes; 2) "targets per route", which indicates the mean number of targets

covered by each patrol route; and 3) "coverage", which alludes to the ratio between the number of targets covered by at least one route and the total number of targets (N_t).

Table 3. Some statistics related to the best set of routes discovered by GAPatrol

Best Patrol Route	Distance	Targets/Route	Coverage	Dangerous	Intermediate	Novice
Scenario 1	115,9	5,1	21 (51.2%)	0	0	15
Scenario 2	111.5	5.5	20 (48.8%)	1	2	12

Table 4. Percentage of targets covered by the GAPatrol best set of routes

Best Patrol Route	Percentage %						
Type	Squares	Drugstores	Lottery	Gas Station	Banks	Mall	Total Coverage
Scenario 1	19.4	22.6	22.6	16.1	12.9	6.4	51.2
Scenario 2	22.3	18.2	22.3	18.2	15.0	0	48.8

For our purposes, we have defined, as the length of a patrol route, the sum of the Euclidean distances between its pairs of contiguous points—each unity of measure corresponds roughly to 100 meters. This metric provides us with the average time spent by a police officer while moving between two adjacent patrolled targets. It is interesting to mention that the mean value of the "distance" parameter for the sets of routes produced in the very first population of the GAPatrol evolutionary engine was in the magnitude of 500, which means that, during the search for the best set of routes, the engine recognized that smaller routes were oftentimes more effective. This is corroborated by looking at the mean number of "targets per route", which was around five for the best set of GAPatrol routes. Actually, some of the produced routes had the smallest possible length, i.e. zero, meaning that they involved only one target. Such a target was perceived by GAPatrol as a candidate to a serious hotspot, and so it allocated one police team to constantly patrol around it. Another strategy discovered by GAPatrol for reducing the lengths of the routes was to allow them to overlap, i.e. to share common points of surveillance. At last, by observing the percentage of total targets covered by the sets of routes, such index was nearly 50% in both scenarios; however, the percentage of points targeted by the novice experts that were covered by the routes was much higher, which indicates that GAPatrol concentrated its efforts mostly on coping with this sort of criminals.

Indeed, by looking again at Table 3, we may notice that none of the criminals could upgrade his/her personality profile in the first scenario, while quite the same phenomenon was observed in the second scenario (only one criminal succeeded in becoming an expert). This contrasted greatly with what had been monitored in the first generations of the evolutionary engine. By referring now to Table 4, one can observe that the types of targets covered by the best set of routes in the first scenario comply with the following distribution: drugstores (22.6%), squares (19.4%), and lotteries (22.6%). In total, 64.5% of the covered targets were hotspots due to novice criminals. Nearly the same percentage was achieved for the second scenario, viz. 62.8%. In sum, all these numbers suggest that the evolutionary approach was very successful in deciphering (i.e., learning) the criminal's behavior model adopted in our experiments. Coping well with the novice criminals was a very operational strategy as

it prevented the appearance of dangerous criminals and implied more effective time spent in patrolling at the targets. As a good side-effect, such strategy would lead to less expenditure in terms of consumable resources (e.g., fuel for patrol cars), although such issue was not put to be directly optimized in the adaptation process.

Finally, it is pertinent to report the results achieved by the human expert. For the first scenario, after some executions of the simulation tool, the expert could realize that the novice criminals were much more inclined to commit their crimes in the vicinities of their points of appearance. That is, that the number of frequently-targeted points was indeed low. Consequently, it was easy for the expert to spot those points and then to allocate the teams to patrol mostly around them. The lengths of the designed routes were indeed very short, covering at most two points. Three out of six suggested routes were formed only by one fixed patrol point in hotspots. In this scenario, the number of crimes prevented by the expert's routes was higher than those prevented by the GAPatrol set of routes. In total, the overall numbers of crimes committed were 80 and 170, respectively. On the other hand, for the second scenario, the expert could not keep up well with its complexity—the amount of targeted points increased significantly. In this case, while the GAPatrol best set of routes achieved the mark of 445 as total number of crimes, the expert's designed routes could achieve only 550 for such index. This result suggests that the strategy of creating small routes per se (adopted by the expert as the only criterion in both cases) does not necessarily imply good performance in all circumstances. Actually, it needs to be combined with other artifices. GAPatrol seems to have recognized such aspect.

4.2 Discussion

The results of the experiments have shown that the GAPatrol approach was effective in detecting the behavior adopted by the criminals. Police patrol routes privileged the hotspots that have emerged from the induced criminal behavior. Notice that this good performance was reached without the introduction of previous human knowledge into GAPatrol. By trying to answer now the two motivation questions made at the beginning of this article, we should point out first that the strategy "adopted" by GAPatrol was to simplify the routes along the evolution process, resorting to heuristic expedients to optimize these routes. Secondly, it was possible to observe that, even though in simple scenarios the human specialist could, in general, outperform the evolutionary approach, in more complex scenarios, the manual configuration of routes was not trivial at all. For that situation, GAPatrol benefited from the artifice of selecting routes that share one or more points in common, forcing the police teams to take sequential turns in such hotspot candidates.

From this discussion, it is to conclude that, when the environment complexity increases, the crime patterns become harder to identify, and the human task of route configuration also becomes more difficult. In our approach, as the problem is dealt with systematically, the requirements imposed by more complex environments will only imply more time to convergence to the (quasi-) optimal solution; typically, this characterizes as an efficiency issue that could be alleviated through the employment of scalable computing infrastructures such as grids. Essentially, the results suggest that, when coping with real-life complex urban scenarios, where the resources are limited and hotspots are distributed geographically, the configurations generated by GAPatrol would have a high chance to outperform those produced manually by human experts.

5 Conclusion and Future Work

In this paper, we have presented an evolutionary multiagent approach, namely GAPatrol, for coping with the ODPR problem. Overall, the achieved results corroborate our earlier perspective [5] that the combination of EAs and MAS [8] comes to be an effective methodology for dealing with non-trivial complex problems, such as those related to the public-safety domain. In the future, we expect that GAPatrol be deployed as a mature auxiliary tool for assisting police tactical managers while devising their preventive patrolling strategies. To this end, a more thorough, systematic analysis, considering alternative simulation scenarios, is underway. Particular aspects under current investigation involve (a) the automatic configuration of the time parcel spent by each police team in each target covered by its associated patrol route; and (b) the possibility of having criminals with capabilities to learn how to better spot their points of attack.

Acknowledgements. The third author has been sponsored by CNPq/Funcap under DCR grant # 23661-04.

References

[1] A. Almeida, G.L. Ramalho, H.P. Santana, P. Tedesco, T.R. Menezes, V. Corruble, and Y. Chevaleyre. Recent advances on multiagent patrolling, In: Bazzan, A.L.C. and Labidi, S., Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, LNAI 3171, pp. 474-483, Springer, Berlin, 2004.

[2] T. Bäck, D. Fogel, and Z. Michalewicz. The Handbook of Evolutionary Computation, Oxford University Press, New York, 1997.

[3] N. Collier. Repast: An extensible framework for agent simulation. Available in http://repast.sourceforge.net, 2003.

[4] A. Machado, G.L. Ramalho, J.-D. Zucker, and A. Drogoul. Multi-agent patrolling: An empirical analysis of alternative architectures, in [6], pp. 155-170.

[5] D. Reis, A. Melo, A.L.V. Coelho, and V. Furtado. Towards Optimal Police Patrol Routes with Genetic Algorithms, in: Mehrotra, S.; Zeng, D.D.; Chen, H.; Thuraisingham, B.; Wang, F.-Y. (Eds.), Intelligence and Security Informatics: IEEE International Conference on Intelligence and Security Informatics (ISI 2006), pp. 485-491, LNCS 3975, Springer, Berlin, 2006.

[6] J.S. Sichman, F. Bousquet, and P. Davidsson (Eds.). Multi-Agent-Based Simulation II, Third International Workshop (MABS 2002), LNAI 2581, Springer, Berlin, 2003.

[7] L. Sherman, P. Gartin, and M. Buerger. Hot spots of predatory crime: Routine activities and the criminology of place, Criminology 27:27-56, 1989.

[8] R. Smith and C. Bonacina. Evolutionary computation as a paradigm for engineering emergent behaviour in multi-agent systems, in: Plekhanova, V. (Ed.), Intelligent Agent Software Engineering, pp. 118-136, Idea Group, 2002.

[9] G. Weiss (Ed.). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press, 1999.

[10] P.A. Winoto. Simulation of the market for offenses in multiagent systems: Is zero crime rates attainable?, in [6], pp. 181-194.

Learning by Knowledge Sharing in Autonomous Intelligent Systems

Ramón García-Martínez, Daniel Borrajo, Pablo Maceri, and Paola Britos

Software and Knowledge Engineering Center, Graduate School,
Buenos Aires Institute of Technology, Argentina
Departamento de Informática, Universidad Carlos III de Madrid, Spain
Intelligent Systems Laboratory. School of Engineering,
University of Buenos Aires, Argentina
rgm@itba.edu.ar

Abstract. Very few learning systems applied to problem solving have focused on learning operator definitions from the interaction with a completely unknown environment. In order to achieve better learning convergence, several agents that learn separately are allowed to interchange each learned set of planning operators. Learning is achieved by establishing plans, executing those plans in the environment, analyzing the results of the execution, and combining new evidence with prior evidence. Operators are generated incrementally by combining rote learning, induction, and a variant of reinforcement learning. The results show how allowing the communication among individual learning (and planning) agents provides a much better percentage of successful plans, plus an improved convergence rate than the individual agents alone.

1 Introduction

Given unknown environments, real autonomous systems must generate theories of how their environment reacts to their actions, and how the actions affect the environment. Usually, these learned theories are partial, incomplete and incorrect, but they can be used to plan, to further modify those theories, or to create new ones. Previous work on machine learning applied to problem solving has mainly focused on learning knowledge whose goal was to improve the efficiency of the problem solving task [Borrajo and Veloso, 1997; Laird et al., 1986; Minton, 1988; Veloso, 1994]. There is also a current interest in learning state transition probabilities in the context of reinforcement learning [Sutton, 1990; Watkins and Dayan, 1992]. However, few researchers have approached the generalized operators acquisition problem [Carbonell and Gil, 1990; Wang, 1996], described as techniques for automatically acquiring generalized descriptions of a domain theory. This issue is crucial when dealing with systems that must *autonomously* adapt to an unknown and dynamic environment.

LOPE (Learning by Observation in Planning Environments) is an agent architecture that integrates planning, learning, and execution in a closed loop, showing an autonomous intelligent behavior [García-Martínez and Borrajo, 1997, 2000]. Learning planning operators (what we will call operators, is also referred to as action models within the reinforcement learning community) is achieved by observing

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 128 – 137, 2006.

the consequences of executing planned actions in the environment. In order to speed up the convergence, heuristic generalizations of the observations have been used. Also, probability distribution estimators have been introduced to handle the contradictions among the generated planning operators. In our previous work, we presented a single agent architecture. Here, we will concentrate on the multiple agents behavior. More concretely, we present the learning mechanism, generalizing the one presented in those papers, and extending it by demonstrating how knowledge may be shared among many agents. The results show how the learning mechanism, outperforms the behavior of the base planner with respect to the production of successful plans (plans that achieve self-proposed goals). But, more importantly, they also show how the interaction with other learning agents greatly improves learning convergence and successful behavior. Section 2 describes the general architecture of the LOPE agents. Section 3 defines the representation that will be used in the paper for situations, observations and planning operators. Section 4 presents the learning model and its components (high level learning algorithm and heuristic generalization of operators). Section 5 deals with how agents share learned knowledge. Section 6 present the experiments and their results. Section 7 compares our approach with related work. Finally, section 8 draws some conclusions.

2 General Description

One of the main objectives of each LOPE agent is to autonomously learn operators (action models) that predict the effects of actions in the environment by observing the consequences of those actions. In order to learn those descriptions, it is able to plan for achieving self-proposed goals, execute the plans, find out incorrect or correct behavior, and learn from the interaction with the environment and other agents. Each agent receives perceptions from the environment, called *situations*, applies actions, and learns from its interaction with the outside world (environment and other agents). At the beginning, the agent perceives the initial situation, and selects a random action to execute in the environment. Then, it loops by executing an action, perceiving the resulting situation and utility of the situation (a classical reward from the environment, further explained in section 3), learning from observing the effect of applying the action in the environment, and planning for further interactions with the environment when the previous plan has finished its execution, or the system observes a mismatch between the predicted situation by the agent's operators and the situation it perceived from the environment. The planner will not be described in this paper. Basically, it does a backward chaining search from the initial situation (goal) of the operator with the highest utility in order to find a sequence of operators that will lead from the current state to that goal. If it succeeds, and the probability of its success is greater than a given bound, it executes the plan. If not, it selects the next highest utility operator and searches for a plan. This process loops until it finds a plan for any high utility operator. More details on how the planner works can be found in [García-Martínez and Borrajo, 1997, 2000]. Figure 1 shows an schematic view of the architecture, where there can be n LOPE agents. Each of the agents receives as input: perceptions from the environment (situation and utilities); set of actions that it

can perform; and operators learned by other agents. The output of each agent is a sequence of actions over time (for the environment), and, regularly the set of operators that it learned (for the other agents).

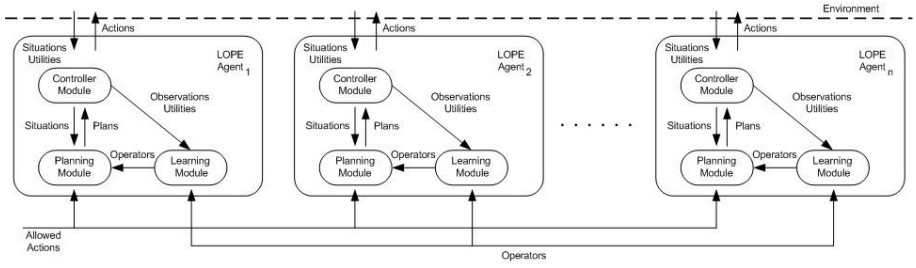


Fig. 1. Architecture of a group of LOPE agents

3 Representation

The autonomous agent type of world that we used for the experiments (robotic tasks) were two-dimensional grids, where each position within a grid can either contain obstacles, energy points, or be empty. For LOPE, as for many other systems, there is a difference between the world states, common to classical planning, and the observations it perceives. While classical planners are mainly interested in the high-level descriptions of the states (e.g. on(A,B) in the blocks world), LOPE builds its operators based on the perceptions it receives from its sensors (the word sensors refer to the generic idea of input, so this is applicable to non robotic domains); its "states" are the inputs it receives from its sensory system. Any post-processing of its inputs in order to translate them into high-level descriptions can be done without affecting the overall behavior. Because of the natural limitations of the sensory system, the agents map different states of the environment in to a single one, so our system manages noisy domains and hidden states. Previous work of one of the authors developed early versions of the learning mechanism. The representation was based on the model proposed in [Fritz *et al*., 1989], in which an observation (also called experience unit) had the following structure: [Initial Situation, Action, Final Situation]; where initial and final situations are lists of propositions that can be either true or false. In [Fritz *et al*., 1989], observations were directly used as planning operators. In [García-Martínez and Borrajo, 1997, 2000], while the concept of an observation does not change, the representation of operators is extended, by the addition of features that allow the system to determine their planning/execution acceptability. The proposed planning operator model has the structure and meaning described in Table 1, where *C*, *F*, *action*, and *U* are domain-dependent and have the following meaning: *C* and *F* describe Initial Situation and Final Situation through a list of propositions (p-list) that can be preceded by the ¬ symbol, denoting negation, if a proposition does not appear on that list, it is assumed that its value does not matter; *action*: can be any of the set of allowed actions that each agent can perform, for instance, in a robotic domain, it could be "go" and "turn" and *U* : is a function that measures how useful the current situation is for the agent, and refers implicitly to the distance to the agent's goal

(similar concept to the reward in reinforcement learning [Watkins and Daya, 1992]). This function could be changed to allow different behaviors, each one depending on their specific goals. The parameters P and K allow the architecture to decrease the effect of noise in the sensors or in the environment, and hidden state problems.

4 Learning Planning Operators

We will first define when operators and observations are similar, equal or one confirms another. Then we will present the high level learning algorithm and the heuristic generalization of operators.

Given two operators $O_1 = [C_1, A_1, F_1, P_1, K_1, U_1]$ and $O2 = [C_2, A_2, F_2, P_2, K_2, U_2]$, and an observation $o = [S_i, A, S_f]$, we say that: (i) two operators are **similar** if $C_1 = C_2$ and $A_1 = A_2$, (ii) two operators are **equal** if $C_1 = C_2$, $A_1 = A_2$, and $F_1 = F_2$, (iii) observation is **similar** to the operator O_1 if $S_i \subseteq C_1$ and $A = A_1$, (iv) observation **confirms** the operator O_1 if $S_i \subseteq C_1$, $A = A_1$, and $S_f \subseteq F_1$. Here, the predicate $\subseteq$ tests whether a list of propositions subsumes another list, and the predicate = tests whether a list of propositions is equal to another.

For presenting the **high level learning algorithm** suppose a situation S_i is perceived by the system, and there exists a set of operators, φ, such that each operator is of the form $O_i = [C,A,F,P,K,U]$. If the system applies the action A, arriving at a situation S_f, the learning method processes this new observation by the algorithm shown in Table 2. When a new observation arrives at the learning module, it checks if a similar operator exists. If it is *similar*, it checks to see if the observation confirms the operator. Then, it rewards all such operators and punishes *similar* ones. If a *similar* operator exists, but there is none that is *confirmed by the observation, it* creates a new operator, punishes *similar* operators to the new one, and generalizes those *similar* operators. The operators generated by the generalization procedure reward *equal* operators and punish *similar* ones. If it does not find a *similar* operator for the input observation, it creates a new one. *Punishing* operators means incrementing in a given quantity *r* the number of times that the pair (condition, action) of *similar* operators to O has been observed. As the algorithm shows, punishment not only occurs when observations are made, but also when new generated by the heuristic generalization. This is so because K really accounts for the number of times that similar operators have been generated or seen. Similarly, *rewarding* operators means incrementing in *r* the P and K of a successful operator. The parameter *r* usually is equal to one, but on Section 5 different values are used for integrating operators of several agents with different Ks. The terms *punish* and *reward* have been borrowed from the field of biological reinforcement rather than from reinforcement learning. The heuristic-generalization algorithm generates a set of new operators according to the generalization heuristics, which are incorporated into the set of planning operators. Since the number of operators that are created can potentially slow down the performance of the learning and planning modules, the system forgets operators with a very low quotient P/K, given enough observations have been made.

The **heuristic generalization of operators** is based on the heuristics defined in [Hayes-Roth, 1983] and [Salzberg, 1985]. For the following discussion, suppose that the new observation is described by (S_i, A, S_f), the domain operator is described by

[C, A, F, P, K, U], and the new generalized observation (m) is [C_m, A_m, F_m, U_m]. In order to apply a heuristic, there had to be a fault in using the corresponding operator in the observed initial and resulting situations.

Hayes-Roth [1983] proposed the following set of heuristics for revising a faulty (buggy) theory: *Retraction* generalizes an operator predicted situation so that it is consistent with the new observation, if $S_f \not\subset F$, then m = [C, A, F', U] where F' is a generalization of F and S_f; *Exclusion* restricts the conditions of the operator, so that it does not apply in the observed situation again, given that $S_i \subseteq C$ and $S_f \not\subset F$, then m = [C', A, F, U], where C' is built by selecting a proposition that does not belong to C (does not matter) and change it to the negation of what appears in the observation; *Inclusion*: generalizes the operator conditions, so that it will later apply in the observed situation. If $S_f \subseteq F$ and $S_i \not\subset C$, then m = [C', A, F, U], where C' is a generalization of C and S_i. It is the equivalent to retraction, but applied to the conditions of the operator.

The following Salzberg (1985) heuristics are used to correct prediction violations: *Inusuality* restricts the conditions of an operator, so that it will not longer apply to the observed initial situation, if $S_i \subseteq$ and $S_f \not\subset F$, then m = [C', A, F, U], where C' is a specialization of C, adding all propositions not appearing in C (does not matter) by the negation of their value in S_i, differs from Hayes-Roth exclusion, in that it adds all propositions to C; *Conservationism*: it is a meta-heuristic that selects the generalization heuristic (from the Salzberg ones), that proposes less modifications in the conditions of an operator; *Simplicity*: it is a generalization of the Hayes-Roth inclusion heuristics; *Adjustment*: when the P/K ratio of an operator falls below a given threshold, it is very unlikely that the operator will correctly predict any situation, if it is a generalization of a set of operators (for instance, by application of the simplicity heuristic), this heuristic generates other combinations of those operators that will increase the ratio.

5 Learning by Sharing

Previous work of the authors presented how the integration of this combined learning mechanism with planning and execution allowed the system to improve the ratio of successful plans (sequence of actions that lead to arrive at the locations of the energy points) [García-Martínez and Borrajo, 1997]. In order to improve the learning convergence, and to test the generality of the learned knowledge, we performed experiments in which the system remembers the operators learned in an example grid g1 when planning and learning in other configurations of the grid, g_2 and g_3 (obstacles and energy points in different places). In those experiments we showed how that prior knowledge provided a faster learning convergence than not using it, and the Section 6 shows the results obtained. We decided then to experiment with the inclusion of new agents of the same type, learning and sharing what they learned in the same grid configuration, and testing how that affected the learning and planning behavior. Agents cannot occupy the same position in the grid, and the sensors of one agent consider the other agents as obstacles. Under this framework, each agent continuously learns, plans and executes. However, when they were close to another agent, they were allowed to communicate in order to interchange what they learned, operator

descriptions. We devised two types of knowledge sharing strategies: Complete sharing and Most reliable operator sharing.

In **complete sharing strategy** every pair of agents integrate their respective theories (set of operators) using all operators in the sets. The algorithm is shown in Table 3. For each operator of another agent (a_2), an agent (a_1) looks for similar operators in its theory. If there is no such similar operator, then the a_2's operator is included in the set of operators of a_1. If a similar operator is found, then all such operators are punished with the P of a_2's operators. If there is no equal operator, then it is included in a_1's operators with the K of its similar operators in a1's theory.

Table 1. General description of an operator

Planning Operator: O_i		
Feature	*Description*	*Values*
C	Initial Situation (conditions)	p-list
A	Action	action
F	Final Situation	p-list
P	Times that the operator O_i was succesfully applied (the expected final situation, F, was obtained)	integer
K	Times that the action A was applied to C	integer
U	Utility level reached applying the action to the initial situation, C, of the operator	real 0..1

Table 2. Algorithm that integrates the operators of two agents

Function Complete-sharing (φ_1, φ_2) : φ_1

φ_1: Set of operator of agent 1
φ_2: Set of operator of agent 2

```
Forall Oi ∈ φ2 do
    If exists Oj ∈ φ1 such that COj = COi
  AND AOj = AOi
    Then φ1 := punish-operators(Oj, φ1, POi);
        If any of the similar operators to Oi,
      Ok, is such that FOk = FOi
        Then Ok := reward-operators(Ok,
      POi)
        Else KOi := KOk;
            φ1 := φ1 U {Oi}
    Else φ1 := φ1 U {Oi};
Return φ1
```

Table 3. Algorithm that modifies operators descriptions after having seen a new observation

Function Learning (S_i, A, S_f, U, φ) : φ

S_i: Initial situation of the observation
A: Applied action of the observation
S_f: Observed final situation
U: Observed utility
φ: Set of operator descriptions

```
If exists Oi ∈ φ such that Si ⊆ COi AND
A = AOi Then
  If Sf ⊆ FOi Then
    Forall Oi such that Si ⊆ COi
    AND =AOi AND Sf ⊆ FOi do
      Oi := reward-operator(Oi, 1);
      UOi := max(UOi, U);
      φ := punish-operators(Oi, φ, 1)
    Else On := [Si, A, Sf, 1, KOi + 1, U];
      φ := φ U {On}
      φ := punish-operators(On, φ, 1)
      M := heuristic-generalization(Si,A, Sf,φ);
      Forall M ∈ M do
        If exists Oj∈ φ such that COj=CM
        AND AOj=AM Then
          If FOj = FM Then
            Oj := reward-operator(Oj, 1);
            φ := punish-operators(Oj, φ, 1)
          Else On := [Cm, Am, Fm, 1, KOj + 1,
      UOj];
            φ := φ U [On];
            φ := punish-operators(On, φ, 1)
        Else φ := φ U {[Cm, Am, Fm, 1, 1, Um]};
Else φ := φ U {[Si, A, Sf, 1, 1, U]}
Return φ
```

In **most reliable operator sharing strategy** every time two agents share their knowledge, only the most liable operators are share (the ones that maximize the

quotient P/K). The only difference with the prior algorithm is that instead of providing it as input with φ_2, the algorithm is called with the set of most liable operators. This set is computed by selecting from each set of similar operators of an agent, the one with maximum P/K.

6 Experiments and Results

We performed several experiments in previous papers to test the behavior of LOPE (García-Martínez and Borrajo, 1997). In order to test the effect of sharing the knowledge among the agents, we performed new experiments which we then compared with a summary of the best results of the previous ones. On each experiment, we averaged the results of running 50 tests. In each test, the initial setup (environment-grid and initial position of the agent) was randomly selected, and each LOPE agent performed 500 cycles of learning, planning and execution. Grids of 700 x 100 pixels were randomly created consisting of 10-20 randomly situated energy points (goals) and obstacles (10% to 20% of the grid area was covered by them). In the multiple agents setup, the agents shared their knowledge when they were 20 pixels away from another. We compare here seven experiments: *SG*: a single LOPE agent learning in a single grid, in which operators are generalized; *SP*: a single LOPE agent learning in a single grid where a probability estimator is assigned to each operator. This estimator is the quotation P/K of each learned operator, also, it is used to assign a confidence to the generated plans, so that plans with low confidence are discarded. The decisions of the agent are based on sensory input only when there is no plan on execution. We have shown previously that the P/K of similar operators follows a multinomial distribution of probability and that is an unbiased estimator of the probability. Also, when an exact theory of the domain exists, the operators that have been built applying the learning mechanism based on observations convergence to the exact ones; *MC*: a set of LOPE agents (we used two for these experiments) learning at the same time in the same grid configuration with the complete sharing strategy; *SGP*: a single LOPE agent learning in a single grid, in which operators are generalized, and a probability estimator is assigned to each operator to assign a confidence to the generated plans, so that plans with low confidence are discarded; *MCG*: a set of LOPE agents learning at the same time in the same grid with the complete sharing strategy and in which operators are generalized; *MCP*: a set of LOPE agents learning at the same time in the same grid configuration with the complete sharing strategy, and where a probability estimator is assigned to each operator to assign a confidence to the generated plans, so that plans with low confidence are discarded; *MCGP*: a set of LOPE agents learning at the same time in the same grid with the complete sharing strategy, in this grid the operators are generalized, and a probability estimator is assigned to each operator to assign a confidence to the generated plans, so that plans with low confidence are discarded.

We used the percentage of successful plans when comparing these versions of the system, and the results of the experiments are shown in Figure 2. First, these results clearly show that the combination of generalization and probability estimation (SGP) outperforms the system using only generalization (SG). Besides, using probability estimation in two agents with complete sharing (MCP) improves

in convergence rate with respect to the SP case, but it worse in the long run, since it converges to a lower percentage rate than the MCP case. It means that sharing knowledge among agents at the beginning is better than using only one agent. Other results show that the use of generalization in two agents with complete sharing (MCG) is worse without using generalization (MC) and even is worse when one agent use generalized operators (SG).

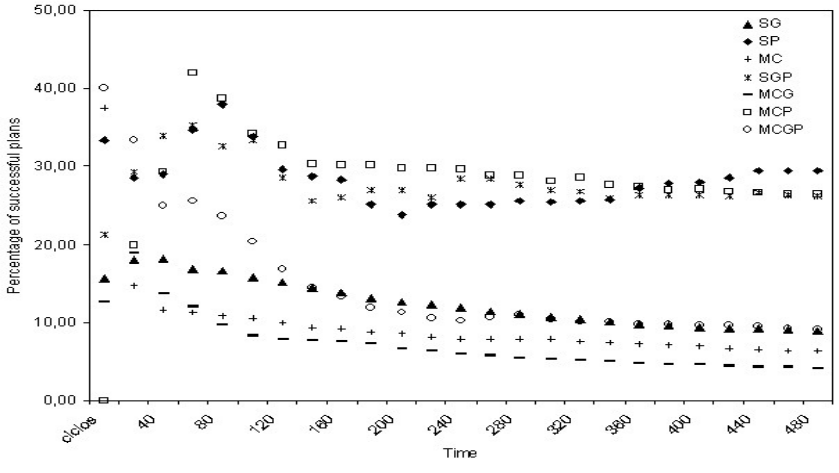


Fig. 2. Results of the experiments

7 Related Work

The closest related work is the one on reinforcement learning techniques within the Markov Decision Processes (MDP) paradigm (Mahavedan and Connell, 1992; Sutton, 1990; Watkins and Dayan, 1992). Also, current techniques that deal with Partially Observable Markov Decision Processes (POMDP) are very close to this approach [Kaebling *et al*., 1998]. Usually, they integrate reinforcement learning, planning and reacting based on approximated dynamic programming. It differs from our work in the fact that the reinforcement procedure is local to an operator, while, in our case, the reinforcement of an operator explicitly implies the punishment of similar ones (global reinforcement). The second difference refers to the fact that we use *symbolically generalized states*, instead of instantiated states (as most other work in reinforcement learning), or non-symbolically based generalized states (such as neural networks [Lin, 1995]. In fact, our approach can be viewed as a method for producing a generalized Q table using global reinforcement. Similar approaches, group sets of similar states and/or actions on big state/action spaces [Boutilier *et al*., 1995]. Most of this work uses different representation schemas, such as belief networks. A third difference lies on the type of planning scheme for which it is used. While reinforcement learning has been usually applied for more reactive planning (with some exceptions), our approach lies closer to the classical planning approach (plans are generated in a search-based fashion and later monitored for divergences between predicted and observed states) [García-Martínez and Borrajo, 1997; 2000]. A fourth difference with most work on

reinforcement learning lies in the fact that we do not deal with the temporal credit assignment problem. Each learning episode is handled independently of what happened before. Within this classical reinforcement learning framework, the work by Tan [1993] could be considered a predecessor of our work. He explores the cooperation among agents by sharing instantaneous information (perceptions, actions or rewards), sequences of perception-action-reward, and learned policies. The GINKO system [Barbehenn and Hutchinson, 1991], the LIVE system [Shen, 1993], and the work of Safra and Tennenholtz [1994] also integrate perception, action and learning. They differ from the proposed architecture in the fact that they do not take into account reinforcement nor heuristic-based refinement of operators. Chrisitansen [1992] also addresses the problem of learning operators (task theories) in a robotic domain. However, in his work there is no revision process as our heuristic-based refinement process. OBSERVER [Wang, 1996] uses an incremental approach for operators revision, where operators involve during the execution of the system. However, there is no memory of past versions of the operators as in LOPE. Also, OBSERVER uses predicate logic for representation, since its goal is to perform classical high-level planning. Our approach uses a representation that is close to the real inputs and outputs of system, with that intermediate type of planning between high-level and reactive planning. Other integrated planning and learning systems for robotic tasks are [Bennet and DeJong, 1996] and [Klingspor *et al*., 1996]. The first one deals with the concept of *permissiveness*, that defines qualitative behavior for the operators. The second one uses Inductive Logic Programming for learning the operators of the domain by doing a transformation from the sensor data into predicate logic. They both differ from our approach in that they need some type of prior background knowledge, either a predefined domain theory in the form of initial operators, or external instruction and knowledge on how to perform the transformation.

8 Conclusions

In this paper, we have presented an architecture that learns a model of its environment by observing the effects of performing actions on it. The LOPE agents autonomously interact with their environment and with other agents with the objective of learning operators that predict, with a given probability estimator, the resulting situation of applying an action to another situation. Two types of knowledge sharing strategies among the agents have been presented: sharing of all the acquired knowledge (operators), and sharing of only the best operator of different sets of operators. The results show that sharing the learned knowledge can greatly help an autonomous system to acquire a theory description that models the environment, thus achieving a high percentage of successful plans, and also improving the convergence rate for obtaining a successful theory. An important issue when allowing sharing of operators among agents, is related to the differences on their sensors, which causes different ways of perceiving the world, and, therefore, different biases towards the generation of operators. We have not yet studied this effect, although one possible way of solving it could be by learning other agents biases, in order to perform a more informed sharing of knowledge. With respect to the scalability of the approach, we are now performing experiments in a much more complex, noisy, with hidden states, and multi-agent domain, such as the Robosoccer. We believe that through the use of

the probabilities estimations, and the heuristic generalization of operators, we will be able to cope with the complexity of that domain.

References

Barbehenn, M. and Hutchinson, S. (1991). An integrated architecture for learning and planning in robotic domains. *Sigart Butlletin*, 2(4), 29-33.

Bennet, S. W. and DeJong, G. (1996). Real world robotics: Learning to plan for a robust execution. *Machine Learning*, 23, 121-162.

Borrajo, D. and Veloso, M. (1997). Lazy incremental learning of control knowledge for efficiently obtaining quality plans. *AI Review Journal*, 11, 371-405.

Boutilier, C., Dearden, R., and Goldszmidt, M. (1995). Exploiting structure in policy construction. *Proc. 14th Int. Joint Conf. on AI:* 1104-1111, Morgan Kaufmann.

Carbonell, J. G. and Gil, Y. (1990). Learning by experimentation: The operator refinement method. In (Michalski and Kodratoff, eds) *Machine Learning: An AI Approach, Vol. III* (pp. 191-213). San Francisco: Morgan Kaufmann.

Christiansen, A. (1992). *Automatic Acquisition of Task Theories for Robotic Manipulation*. PhD thesis, School of Computes Science, Carnegie Mellon University.

Fritz, W., García-Martínez, R., Blanqué, J., Rama, A., Adobbati, R., and Sarno, M. (1989). The autonomous intelligent system. *Robotics and Autonomous Systems*, 5, 109-125.

García-Martínez, R. and Borrajo, D. (1997). Planning, learning, and executing in autonomous systems. *Lecture Notes in Artificial Intelligence*, 1348, 208-220.

García Martínez, R. y Borrajo, D. (2000). An Integrated Approach of Learning, Planning and Executing. *Journal of Intelligent and Robotic Systems*, 29, 47-78.

Hayes-Roth, F. (1983). Using proofs and refutations to learn from experience. In (Michalski, Carbonell, and Mitchell eds) *Machine Learning, An AI Approach* (pp. 221-240). Palo Alto: Tioga Press.

Kaebling, L., Littman, M. and Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 10, 99-134.

Klingspor, V., Morik, K., and Rieger, A. (1996). Learning concepts from sensor data of a mobile robot. *Machine Learning*, 23, 305-332.

Lin, L. (1995). Reinforcement Learning of Non-Markov Decision Processes. *Artificial Intelligence*, 73, 271-306.

Mahavedan, S. and Connell, J. (1992). Automatic programming of behavior-based robots using reinforce-ment learning. *Artificial Intelligence*, 55, 311-365.

Safra, S. and Tennenholtz, M. (1994). On planning while learning. *JAIR*, 2, 111-129.

Salzberg, S. (1985). Heuristics for inductive learning. *Proc. 9th International Joint Conference on AI*, pages 603-609, Los Angeles, CA.

Shen, W. (1993). Discovery as autonomous learning from environment. *Machine Learning*, 12, 143-165.

Sutton, R. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proc. 7th Int. Conf. on ML*: 216-224. Kaufmann.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. *Proc. 10th International Conference on ML* (pp. 330-337), Amherst: Morgan Kaufman.

Wang, X. (1996). *Planning while learning operators*. PhD thesis, School of Computes Science, Carnegie Mellon University.

Formal Analysis of a Probabilistic Knowledge Communication Framework

João Carlos Gluz[1], Rosa Maria Viccari[2], Cecilia Dias Flores[2], and Louise Seixas[2]

[1] PIPCA/UNISINOS, 93022-000, São Leopoldo, RS, Brazil
[2] UFRGS, 95501-970, Porto Alegre, RS, Brazil,
jcgluz@unisinos.br, {rosa, dflores}@inf.ufrgs.br,
seixas@farmacia.ufrgs.br

Abstract. This paper introduces a new formal model, which generalizes current agent communication theories (basically the FIPA version of these theories) to handle probabilistic knowledge communication. Several questions about communication of probabilistic knowledge are discussed in the light of current theories of agent communication and it is argued that exists a semantic gap between these theories and research areas related to probabilistic knowledge representation and communication. This gap creates serious theoretical problems if agents that reason probabilistically try to use communication framework provided by these theories. To diminish this gap it is proposed a modal probabilistic logic and a new communication framework composed of communication principles and acts for probabilistic knowledge communication.

Keywords: Artificial Intelligence, Multiagent Systems, Agent Communication Languages, Probabilistic Logic, Probabilistic Knowledge Communication.

1 Introduction

This paper will present a theoretical study about which kind of meaning can be assigned to the communication of probabilistic knowledge between agents in Multiagent Systems (MAS), at least when current theories for agent communication are considered. The work starts in section 2, presenting several considerations showing that exists a semantic gap between current agent communication theories and research areas related to probabilistic knowledge representation and communication. This gap creates very serious theoretical problems if the designer of agents that reason probabilistically tries to use the communication framework provided by these theories to model and implement all agent's communication tasks.

To diminish this gap a new formal model is proposed in section 3. This model will generalizes the formal model, used in FIPA agent communication standards [6], to handle probabilistic knowledge communication. A new probabilistic logic, called *SLP*, is defined as the basis for the new model. The *SLP* logic is compatible with the *SL* logic used as the foundation of FIPA standards in the sense that all valid theories of *SL* are also valid in *SLP*. The axiomatic system of *SLP* is correct. It is complete, if the axiomatic system of *SL* is complete.

Based on *SLP* logic it is proposed, in section 4, a communication framework able to correlate probabilistic reasoning with communication related inference tasks. Two

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 138 – 148, 2006.

new communicative acts would allow agents to communicate probabilistic propositions without having to agree previously on content formats.

This is the main result of the paper. To our knowledge, this is the first work that tries to integrate in a single probabilistic-logical framework two entirely different approaches to understand and model communication. First, we carefully isolated formal axiomatic agency and communication theories used by FIPA and then we specify the minimum set of new axioms to be added to these theories, necessary and sufficient to support assertive probabilistic communicative acts. We keep principles, acts and axioms as simple as possible to be able to easily assess how much we were departing from classical Speech Act theory. A secondary, but interesting, result of the paper is the relative completeness of *SLP* logic. To our knowledge, there is no other axiomatization for an epistemic and temporal modal logic, which allow probabilities for first order modal sentences, and is proved complete.

2 Motivation and Related Work

This work has started with a concrete problem, which was how to model (and implement) the communication tasks of all agents from AMPLIA system [13,8]. We decided to use FIPA standards to do this tasks based on two assumptions: (a) the standards are a good way to ensure MAS knowledge reusability and interoperability; (b) the formal basis of FIPA standards offer an abstract and architecture independent way to model all communication tasks of the system, allowing a high level description of the communication phenomena. However, we have found that it was impossible to meet even most basic communication requirements of AMPLIA using only FIPA standards. All AMPLIA's agents use and communicate probabilistic (bayesian) knowledge, but FIPA standards assigns no meaning to probabilistic knowledge representation or communication.

Of course it is possible to encode all probabilistic knowledge in a special new content format, allowing, for example, that Bayesian Networks (BN) should be encoded in this format and then embedded as contents of FIPA Agent Communication Language (ACL) communicative acts. Encoding is considered here only a syntactic technique and not a full knowledge representation method because axioms, inference rules and implications of Probability Theory are not taken into consideration. This implies that the probabilistic knowledge to be passed as contents of assertive **inform** acts must be encoded in atomic terms of logical formulas. From the communication point of view, it is only necessary that the agent believe that the encoded probabilistic knowledge transported by the act is logically true. Any other meaning related the probabilistic knowledge do not need be known by the agent in respect to communication tasks or in any reasoning related to this tasks.

The approach to encode probabilistic knowledge presents some problems if theoretical or formal aspects of this kind of communication are considered. In the first place, the formal semantics of FIPA ACL is based on axiomatic logical theories of intention and communication [4,5,11,12]. Besides particular pre and pos-conditions (expressed as logical axioms) for some act, these theories define clearly when the act should be emitted, what are the intentions of the sender agent, which effects this act should cause in the receiver agent and so on. The knowledge transported in these acts

are only logical propositions, but this propositions are related to internal beliefs, intentions and choices of the agents and must be used in reasoning process that will decides when to emit some act or how the act received should be understood. This imply that even if there is some probabilistic knowledge encoded in the contents of a communicative act, then this knowledge *cannot be used* in any internal reasoning process related to communication tasks, because formal model and theories that fundament this reasoning (at least in FIPA standards) are purely logical and do not allow reasoning about probabilities. The second question arises from epistemological and linguistic considerations, when we consider agents that can reason probabilistically. If some agent uses subjective (bayesian) reasoning and assign probabilities to his beliefs, then, assuming only basic rationality for this kind of agent, if it has some probabilistic belief and needs to inform this belief to another agent, it will need to be sure that the proper degree of belief be also correctly informed. Some appropriate locus for the transportation of this kind of probability needs to be found in current theories of communication. The problem is that the Speech Act Theory of Searle and Grice, which provides the epistemological and linguistic basis for formal communication theories, simply do not consider the possibility of agents to communicate knowledge of probabilistic nature because the most basic semantic unit of knowledge considered is a logical proposition. Consequently, main formal theories of communication and intention [4,5,11,12] adopted this point of view and do not consider the possibility of probabilistic knowledge communication.

Together both questions create a interesting dilemma: if an agent use probabilistic reasoning and need to inform some probabilistic belief to another agent it will have serious problems to do this task, because current linguistic theories say that there is no means to accomplish it (according to these theories there is no *locus* to communicate probabilities). Even if it is possible to send this probabilistic knowledge there is no way to consider this knowledge when reasoning about communication tasks.

The problems expressed above are not addressed in recent research literature about ACLs [3]. Research in this area and related areas of agent societies and social interaction is more focused in the study about logical aspects of social institutions, including trust relationship, intentional semantics for social interaction and similar concepts, but not in checking the role of probabilities in these concepts. A similar situation also occurs in the research area of probabilistic knowledge representation for MAS. Main papers in those areas are focused on the question of how to communicate and distribute BN probabilistic knowledge between agents [14], keeping the inference processes consistent, efficient and epistemologically sound. These pieces of research offer a separate form of knowledge representation and communication not related to ACL research.

Our work intends to start to bridge this gap, by showing how FIPA communication framework can be extended to support probabilistic knowledge in an integrated and compatible way. The base of our work is a *probabilistic modal logic*. In terms of AI, probabilistic logics were first described by Nilsson [10]. The initial work of Nilsson was profoundly extended by the works of Halpern [9], Abadi [1] and Bacchus [2] mainly related to doxastic/epistemic probabilistic modal logics. Due to the nature of the theories of agent communication, that require BDI modal operators, we focused our research only on doxastic probabilistic modal logics, so these works provide the formal analysis tools used in this paper. Currently there is also other active lines of

research on probabilistic logics, mainly based on probabilistic extensions to the CTL* temporal logic from Emerson and Srinavan. These logics, however, are not related to our work because they do not support belief modalities of doxastic/epistemic logics.

3 Formal Model

3.1 SLP Logic

The agent communication framework presented in this paper follows FIPA approach to define agent communication languages and protocols. It is based in a modal probabilistic logic called *SLP*, for *Semantic Language with Probabilities*, which is an extension of the BDI-like modal logic with equality that fundaments FIPA communication standards. The logic used by FIPA is called *SL* and was defined in [11,12] with a possible-worlds model-based semantics. However, in *SL* it is not possible to attribute subjective probabilities (or degrees of belief) to beliefs, so it is not possible to represent or reason about probabilistic knowledge in this logic.

SLP inherits from *SL* the usual operators and quantifiers of the first order predicate logic with equality. It inherits too modal operators to express the beliefs ($\boldsymbol{B}(a,\varphi)$), choices $\boldsymbol{C}(a,\varphi)$, intentions ($\boldsymbol{I}(a,\varphi)$) and absolute uncertainties ($\boldsymbol{U}(a,\varphi)$) of some agent a. It also inherits action expressions that can be connected in series $e_1;e_2;...;e_n$, in alternates $e_1|e_2$ or verified by an agent a $(a,e)?$. Temporal and possibility assertions can be made based on the fact that an action or event has happened ($\boldsymbol{Done}(e, \varphi)$), on the possibility that an action or event may happen ($\boldsymbol{Feasible}(e, \varphi)$) and on which agent is responsible for an action ($\boldsymbol{Agent}(a,e,\varphi)$).

The new elements of *SLP* are defined through the extension of *SL* semantic model and axiomatic system by the incorporation of probabilistic terms and numerical terms and expressions. The probabilistic term $\boldsymbol{BP}(a,\varphi)$ is specific for *SLP* and informs the probability of a proposition φ be true with respect to the beliefs of agent a, that is, it defines the subjective probability assigned to φ by a. For example, $\boldsymbol{BP}(a,\exists(x)(\mathrm{P}(x))\leq\mathbf{1}$ express the fact that the subjective probability assigned by agent a to the possibility that some element of the domain satisfies $\mathrm{P}(x)$ is less than **1**.

The model-based semantics for formulas of *SLP* is defined over a set Φ of symbols for variables, functions, predicates, primitive actions, agents and constants through models M with the following structure:

$$M = <\mathcal{W}, \mathcal{A}gt, \mathcal{E}vt, Obj, \mathcal{B}, \mathcal{C}, \mathcal{E}, \mathcal{AGT}, \sigma, \mathcal{RCF}, \mu>$$

The elements $\mathcal{W}$, $\mathcal{A}gt$, Obj, $\mathcal{E}vt$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{E}$, $\mathcal{AGT}$ and σ are part of the formal model originally defined for *SL* that are used in *SLP* with the same definition. Detailed definition and properties of these elements is presented in [12]. As a brief introduction, the elements $\mathcal{W}$, $\mathcal{A}gt$, Obj, $\mathcal{E}vt$ and $\mathcal{AGT}$ constitute the non-void sets of possible worlds ($\mathcal{W}$), agents ($\mathcal{A}gt$), objects (Obj), primitive events ($\mathcal{E}vt$) and causative agent for primitive events ($\mathcal{AGT}$) of *SLP* models. The elements $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{E}$ define the sets of accessibility relations for beliefs ($\mathcal{B}$), choices ($\mathcal{C}$) and future worlds ($\mathcal{E}$) of *SLP*. The mapping σ denotes a standard first-order logic interpretation that attributes, for each possible world, function and predicate symbol in Φ a correspondent element in $\mathcal{A}gt \cup Obj \cup \mathcal{E}vt$.

The elements μ and $\mathcal{RCF}$ are new elements necessary for the semantics of *SLP*.

Definition 1. *The set μ is a set of mappings $\mu_a: \mathcal{W} \rightarrow \mathcal{RCF}$ that attributes to each agent a a discrete probability distribution function on the set of possible-worlds $\mathcal{W}$. The basic restriction to this set of mappings is that any mapping μ_a must respect the restrictions for any discrete probability function, that is, they must respect the following condition: (a) $\mu_a(\mathcal{W})=1$, (b) $\mu_a(w) \geq 0$, for all $w \in \mathcal{W}$, and (c) $\mu_a(\mathcal{W}_1 \cup \mathcal{W}_2 \cup \mathcal{W}_3 \cup ...) = \mu_a(\mathcal{W}_1) + \mu_a(\mathcal{W}_2) + \mu_a(\mathcal{W}_3) + ...$, for all $\mathcal{W}_i$ such that $\mathcal{W}_i \subseteq \mathcal{W}$ and for all pairs $\mathcal{W}_i$ and $\mathcal{W}_j$, with $i \neq j$, such that $\mathcal{W}_i \cap \mathcal{W}_j = \varnothing$.*

Definition 2. *The symbol $\mathcal{RCF}$ denotes the (up to isomorphism) closed ordered field of real numbers [15]. $\mathcal{RCF}$ it is the domain for the purely numerical formulas of SLP and includes addition (+) and multiplication ($\times$) operations on real numbers, the neutral elements of these operations (0,1), the partial ordering $\leq$ and it satisfies all properties of real-closed ordered fields [15].*

The formal semantics of *SLP* expressions, that are not probabilistic, are identical to the semantics given for *SL* in [12]. The presentation of the semantic for the entire *SLP* logics is out of the scope of present work (it is defined in [7]), however, here we will define the formal semantics of the basic belief relation $\boldsymbol{B}(a,\varphi)$ and of the new probabilistic term $\boldsymbol{BP}(a,\varphi)$, to show the correlation between these two constructions.

Definition 3. *The modal operator $\boldsymbol{B}(a,\varphi)$ expresses the fact that the agent a beliefs that the sentence φ is true in a model M, world w and evaluation function v if and only if φ is true in any world w' which can be reached from w using $\mathcal{B}_a$ the belief accessibility relation for the agent a*:

$$M,w,v \models \boldsymbol{B}(a,\varphi) \textbf{ iff } M,w',v \models \varphi, \qquad \text{for all } w' \text{ such that } w \, \mathcal{B}_a \, w'.$$

Definition 4. *The semantic of the probabilistic term $\boldsymbol{BP}(a,\varphi)$ is the probability estimated by agent a that φ is true. This probability is calculated summing up the distribution function μ_a over the worlds where agent a believe that φ is true*:

$$[\boldsymbol{BP}(a,\varphi)]_{M,w,v} = \quad \mu_a(\{w' \mid w \mathcal{B}_a w' \text{ and } M,w',v \models \varphi\})$$

Besides these definitions, we add two assumptions to the formal model of *SLP*.

Assumption 5. *The following equivalences are valid in SLP*:

$$\boldsymbol{B}(a, \varphi) \Leftrightarrow \boldsymbol{BP}(a, \varphi)=\mathbf{1}$$
$$\boldsymbol{U}(a, \varphi) \Leftrightarrow \boldsymbol{BP}(a, \varphi)=\mathbf{0.5}$$

This assumption states the basic relationship between probabilistic and non-probabilistic (i.e. purely logical) beliefs in *SLP* and between absolute uncertainties of *SL* and probabilistic beliefs.

Assumption 6. *Any formula φ inside $\boldsymbol{BP}(a,\varphi)$ terms must be a sentence (a closed formula) of the logic. Numerical constants or variables cannot be used as arguments of logical predicates (and vice-versa).*

The axiomatic system of *SLP* was built over the axiomatic system of *SL*. It incorporates all axioms and inference rule from *SL*. To support probabilities were added the axiomatic system for the real-closed ordered field of numbers [15] and axioms and inference rules equivalent to Kolmogorov axioms for Probability Theory.

3.2 Properties of SLP Logic

The basic properties of SLP are enunciated in the following propositions.

Proposition 7. *Any valid formula of SL is also a valid formula of SLP and any purely logical valid formula of SLP is a valid formula of SL.* □

The proof of this proposition is not so simple because of assumption 5 which forces that every world with nonzero probability from a M model can be reached by any other world of this model through the $\mathcal{B}$ relation, something that is not required in *SL* (or in other epistemic modal logics). Even so, it was possible to prove in [7], that for any valid model of *SL* exist a valid model of *SLP* and *vice-versa*. This fact implies in the proposition 7.

Proposition 8. *The axiomatic system of SLP is correct.* □

The new axioms and inference rules of *SLP* are derived from the axiomatic theory of probabilities from Kolmogorov and from the axiomatic theory of real-closed ordered fields. In [7] is proved that these axioms are valid formulas and that the new inference rules preserve the satisfiability of their deductions. In conjunction with the assumed correctness of *SL* axiomatic system, this is enough to prove the correctness of the axiomatic system of *SLP*.

In our proposed extension to *SL*, we have taken special care to avoid the problem of undecidability of probabilistic logics described in [1]. We have found a very interesting result, showing that there is a simpler and intuitive set of restrictions, not so strong as the restrictions proposed by Halpern [9] and Bacchus [2], which keep the resulting axiomatic system complete.

Proposition 9. *The axiomatic system of SLP is complete if the axiomatic system of SL is also complete.* □

The proof of completeness of SLP is presented in [7]. The proof is large and complex, however the basic insight that leads to this proof can be presented here. This insight was based on the observation that the main technique used in the incompleteness proof for probabilistic logics made by Abadi and Halpern [1] relied strongly on the fact that one particular variable can be used by formulas inside an probabilistic operator $\boldsymbol{BP}(a,\varphi)$ and by formulas outside this operator. In this case is necessary, for Abadi and Halpern's technique to work, to be possible to construct expressions like $(\forall x)(P(x) \wedge \boldsymbol{BP}(a,Q(x))=r)$, where the same variable x is shared by predicate $P(x)$ that is inside the $\boldsymbol{BP}$ operator and by predicate $Q(x)$ which is not. The consequence is that if this sharing of variables between probabilistic terms and logical formulas it is not allowed in some probabilistic logic, then Abadi and Halpern's incompleteness proof technique will not work for this logic. This, of course, does not imply that the axiomatic system of this logic is complete, but it shows that this should be possible. Indeed, if we do not allow this kind of sharing, as is the case of *SLP* because of

assumption 6, it is possible to generalize the proof techniques used by Halpern [9] to the case of *SLP* and separate the probabilistic and non-probabilistic parts of any valid formula of this logic. This is the basic method employed in [7] to prove the completeness of *SLP*. The details are complex because, different from logics presented in [9], *SLP* allows fully quantified first order modal formulas inside $\boldsymbol{BP}(a,\varphi)$ terms. However, in [7] it was shown that the validity of any formula θ of *SLP* can be reduced to the validity of an equivalent formula $\psi \wedge \pi$, where ψ is a purely logical formula containing no numerical or probabilistic term and π is a purely numerical formula containing no logical predicate/term neither any probabilistic term. There is no probabilistic $\boldsymbol{BP}(a,\varphi)$ term in ψ or in π. This result was proved using a finitary generalization of Halpern techniques presented in [9] that allowed the substitution of all $\boldsymbol{BP}(a,\varphi)$ terms in θ by equivalent universally quantified numerical equations in π. The validity of formula ψ depends entirely on the *SL* axiomatic system, because ψ is a purely logical formula, so it is a formula of *SL* and, if ψ is valid, it must be valid in *SL*. The validity of π depends on the first order axiomatic theory of real closed fields that, because it is a formula formed only by universally quantified first order numerical equations. This is a decidable problem by a well-known result of Tarski [15], that is, if the set of universally quantified numerical equations that form π is a valid set of equations then exist a proof of this in the axiomatic theory of real closed fields.

4 Probabilistic Communication Framework

4.1 Probabilistic Communication Principles

The FIPA ACL semantic depends on several logical axioms that define principles for agency and communication theories (see [11,12] for details). The theory of agency employed by FIPA includes rationality, persistency and consistency principles for beliefs, choices and intentions of agents defined as *SL* axioms and theorems. The theory of communication is formed by several axioms that define communication principles like the belief adjustment, sincerity, pertinence and cooperation principles besides the 5 basic communication properties stated in FIPA ACL specification [6]. These principles are generally sufficient to handle reasoning needs for communication purposes in any rational BDI agent that is FIPA compliant (at least when the sender's agent centered semantics used by FIPA ACL is appropriate for the application or domain in question). In being so, our first principle can be stated as the following assumption.

Assumption 10. *Agents that need to communicate probabilistic knowledge and intend to use FIPA-ACL should also respect the theory of agency and the theory of communication proposed in FIPA standards.*

This assumption is perfectly reasonable because of compatibility between *SL* and *SLP* assured by proposition 7, that implies that any valid theory of *SL* is a valid theory of *SLP*. However, when agents use probabilistic reasoning and need to use this kind of knowledge for communication purposes, then the purely logical theories of agency and communication are not much useful. To handle these situations we propose that

these theories be extended by two new principles that will be able to bridge the gap between purely logical considerations and probabilistic reasoning, in terms of agent's communication decisions. We will propose only a minimum set of new principles, strictly necessary to correlate probabilistic knowledge used by the agent to decision and inference processes related to communication tasks.

One fundamental property of FIPA theory is the principle that assures the agreement between the mental state of some agent and their beliefs [12]. Using this principle is possible to assert propositions like $\boldsymbol{B}(a,\varphi) \leftrightarrow \boldsymbol{B}(a,\boldsymbol{B}(a,\varphi))$ and $\boldsymbol{BP}(a,\varphi)=\mathbf{1}\leftrightarrow\boldsymbol{B}(a,\boldsymbol{BP}(a,\varphi)=\mathbf{1})$, if all propositions and predicate symbols in φ appears in the scope of a modal operator formalizing a mental attitude of agent *a*:

This is an interesting fact but is very limited in the case of probabilistic communication. The principles of FIPA's theory of communication assume that the agent must believe non-probabilistically in some fact, before the communication starts. Therefore, what we need is some principle that will allow us to correlate probabilistic beliefs with non-probabilistic beliefs. This is assured by the following proposition of *SLP*.

Proposition 11. *Principle of Probabilities and Beliefs Agreement: if the subjective probability assigned for an agent to φ is p, then it believes on this fact and vice-versa*:

$$\models \boldsymbol{BP}(a, \varphi)=p \leftrightarrow \boldsymbol{B}(a, \boldsymbol{BP}(a, \varphi)=p\ ) \qquad \square$$

This principle allows agents to put any probabilistic beliefs "inside" epistemic belief operators and then to use any other axioms and theorems of communication or agency theories to make communication related reasoning. The proposition 11 is necessary but is not enough. We need some kind of reason to effectively start some new communicative act. In FIPA this is assured by the principle of belief adjustment [12] that states that if some agent *a* believe in *φ*, believe that is competent in this belief ($\boldsymbol{Comp}(a,\phi)$) and thinks that another agent *b* do not believe in *φ*, then it adopts the intention to make *b* believe in *φ*:

$$\models \boldsymbol{B}(a,\ \varphi \wedge \boldsymbol{B}(b, \neg\varphi) \wedge \boldsymbol{Comp}(a,\ \varphi)) \rightarrow \boldsymbol{I}(a, \boldsymbol{B}(b,\ \varphi))$$

The belief adjustment principle also falls in the same limiting situation of the mental state and belief agreement principle when applied to the probabilistic case. Therefore, we need another principle stated in the following proposition.

Proposition 12. *Principle of Probabilities Adjustment: if some agent a believe that the probability of proposition φ is p, believe that it is competent in this belief and also believe that another agent b have different estimation for the probability of φ, then it should adopt the intention to make agent b also believe that the probability of φ is p*:

$$\models \boldsymbol{BP}(a,\varphi)=p \wedge \boldsymbol{BP}(a,\boldsymbol{BP}(b,\varphi)=p)<\mathbf{1} \wedge \boldsymbol{B}(a,\boldsymbol{Comp}(a,\boldsymbol{BP}(a,\varphi)=p))) \rightarrow \boldsymbol{I}(a, \boldsymbol{BP}(b, \varphi)=p) \qquad \square$$

This principle is derived from belief adjustment principle, using the proposition 11 stated before (see [7] for details). It will have the same function of belief adjustment principle for the probabilistic reasoning case, providing agents with intentions to solve perceived differences between probabilistic beliefs shared by several agents.

4.2 Communicative Acts for Probabilistic Knowledge

Like *SL*, *SLP* also can be used as a content representation language for FIPA-ACL communicative acts. This allows the representation and distribution of probabilistic knowledge like BN between agents using standard assertive (**inform**) acts. However, to do this is necessary to assume a particular structure in the contents of these acts. The assertive acts defined in Speech Act theory (and the equivalent **inform** FIPA-ACL acts) do not assume any particular internal structure in the propositions passed as contents of these acts. So, in the general case of probabilistic communication not seem reasonable to always assume a particular structure in the content of assertive act used to communicate probabilities. To handle this we propose that the strength (or weakness) of the assertive force of some speech act should be measured by a probability. In this way, any kind of propositions can be used as contents of these probabilistic assertive acts, because the (subjective) probability of the proposition will be transmitted as a graduation of the force. This graduation is a numerical coefficient that represents the subjective probability of the proposition (i. e., the graduation of the assertive force is directly related to the belief degree on the proposition). Two new probabilistic communicative acts were defined. They are considered extensions to the FIPA-ACL, creating the *Probabilistic Agent Communication Language* (PACL).

The acts **inform-bp** and **query-bp** acts are defined, respectively, to allow that the information about subjective probabilities of an agent to be shared with other agents and to allow that a given agent could query the degree of belief of another agent. Using the notation employed by FIPA-ACL [6] the formal definition of **inform-bp** is given by the following schema:

$\langle a,$ **inform-bp** $(b, \langle \varphi, p \rangle)\rangle$
FP: $\boldsymbol{BP}(a,\varphi)=p \wedge \boldsymbol{BP}(a, \boldsymbol{BP}(b,\varphi)=p) < \mathbf{1}$
RE: $\boldsymbol{BP}(b,\varphi)=p$

This act informs the probability for some closed formula φ. The feasibility precondition of the act (*FP*) requires only that an agent to believe that the subjective probability of φ is *p* and that another agent *b* has the chance of not believing in this fact. In this case, if the other necessary conditions are fulfilled (see [6]), then the **inform-bp** act will be emitted. The rational effect (*RE*) that is expected with the act emission is that agent *b* also comes to believe that the probability of φ is *p*.

The **query-bp** act was also modeled after an analysis of the **query-if** act, which is its similar when dealing with truth-values. This directive act is used to retrieve the probabilistic information associated to a particular proposition.

5 Applications and Future Works

The new agent communication framework briefly introduced in sections 3 and 4 not only is compatible with current agent communication standards, but also has a solid theoretical foundation. However, to know the applicability of the framework it is necessary to show how it can be used to model the communication of probabilistic knowledge in several kinds of situations.

Returning to the original research motivation, the ability of the framework to model the communication of a complex MAS was shown in [16,8] with the formalization of the communication of the AMPLIA system. This formalization was used to analyze the abstract properties of the negotiation processes that occur in AMPLIA and offered clear design guides for the implementation of the communication behavior of AMPLIA's agents. In [7] was show how any discrete BN can be transformed in some equivalent set of *SLP* formulas, that is, how *SLP* can be used as the content language to express and share probabilistic knowledge between agents, when the knowledge representation format is agreed by all agents. The capability of the new framework to represent complex interaction protocols is also show in [7], were is proposed a set of interaction protocols, specified in AUML, that ensure that global and local consistency of Multiply Sectioned BN (MSBN) can be achieved in a multi-agent scenario.

Several interesting developments can follow our work. A direct possibility it is to check the influence of probabilistic knowledge and reasoning in other types of communicative acts and interaction protocols. Another possibility is to use the logical representation schemes for BN (like the schemes presented in [2] and [7]) as a starting point for the research of shared ontologies for probabilistic knowledge. The considerable research work already done for logical based ontologies, can be applied to this new research.

Acknowledgments. The authors gratefully acknowledge the Brazilian agencies CAPES, CNPq and FAPERGS for the partial support to this research project.

References

1. Abadi, M. and Halpern J.: Decidability and Expressiveness for First-Order Logics of Probability. In: Procs. of IEEE Symp. on Foundations of Computer Science, 30, 1989.
2. Bacchus, F.: Lp, a Logic for Representing and Reasoning with Statistical Knowledge. Computational Intelligence, 6:209-301, 1990.
3. Chaib-Draa, B. and Dignun, F.: Trends in Agent Communication Language. Computational Intelligence v. 2, n. 5. Cambridge, MA: Blackwell Publ., 2002
4. Cohen, P. and Levesque, H.: Rational Interaction as the Basis for Communication, In: P. Cohen, J. Morgan and M. Pollack (eds.). Intentions in Communication. Cambridge, MA: MIT Press, 1990.
5. Cohen, P. and Levesque, H.: Communicative Actions for Artifical Agents. In: Procs. of ICMAS-95. San Francisco. Cambridge: MIT Press, 1995.
6. FIPA. FIPA Communicative Act Library Specification, Std. SC00037J, FIPA, 2002.
7. Gluz, J. C.: Formalization of the Communication of Probabilistic Knowledge in Multiagent Systems: an approach based on Probabilistic Logic (In Portuguese). PhD Thesis. Instituto de Informática, UFRGS, Porto Alegre, 2005.
8. Gluz, J. C., Flores, C. D., Seixas, L., Viccari, R. M.: Formal Aspects of Pedagogical Negotiation in AMPLIA System. In: Procs. of TISE-2005. Santiago, Chile, 2005.
9. Halpern, J. Y.: An Analysis of First-Order Logics of Probability. Artificial Intelligence, 46: 311-350, 1990.
10. Nilsson, N. J.: Probabilistic Logic. Artificial Intelligence, Amsterdan, 28: 71-87, 1986.
11. Sadek, M. D.: Dialogue Acts are Rational Plans. In: Procs. of ESCA/ETRW Workshop on the Structure of Multimodal Dialogue, Maratea, Italy, 1991.

12. M. D. Sadek. A Study in the Logic of Intention. In: Procs. of KR'92, p. 462-473, Cambridge, USA, 1992.
13. Vicari, R. M., Flores, C. D., Seixas, L., Silvestre, A., Ladeira, M., Coelho, H. A.: Multi-Agent Intelligent Environment for Medical Knowledge. Artificial Intelligence in Medicine, 27(3): 335-366, March 2003.
14. Xiang, Y.: A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication. Artificial Intelligence, 87: 295-342, 1996
15. Monk , J. D.: Mathematical Logic. New York: Springer-Verlag, 1976.
16. Gluz, J.C., Flores, C.D., Vicari, R.M.: Formal Aspects of Pedagogical Negotiation in AMPLIA System. To appear in: N. Nedjah, L.M. Mourelle (eds.) Intelligent Educational Machines. Series: Intelligent Systems Engineering Book Series. Springer/Nova, 2006.

Color Image Segmentation Through Unsupervised Gaussian Mixture Models

Antonio Peñalver, Francisco Escolano, and Juan M. Sáez

Robot Vision Group
Alicante University, Spain
a.penalver@umh.es, {sco, jmsaez}@dccia.ua.es

Abstract. In this paper we introduce a novelty EM based algorithm for Gaussian Mixture Models with an unknown number of components. Although the EM (Expectation-Maximization) algorithm yields the maximum likelihood solution it has many problems: (i) it requires a careful initialization of the parameters; (ii) the optimal number of kernels in the mixture may be unknown beforehand. We propose a criterion based on the entropy of the pdf (probability density function) associated to each kernel to measure the quality of a given mixture model, and a modification of the classical EM algorithm to find the optimal number of kernels in the mixture. We apply our algorithm to the unsupervised color image segmentation problem.

1 Introduction

Mixture models, in particular those using Gaussian kernels, has been widely used in the field of statistical pattern recognition. One of the most common methods for fitting mixtures to data is the EM algorithm [6]. However, this algorithm is prone to initialization errors and, in these conditions, it may converge to local maxima of the log-likelihood function. In addition, the algorithm requires that the number of elements (kernels) in the mixture is known beforehand. For a given number of kernels, the EM algorithm yields a maximum-likelihood solution but this does not ensure that pdf of the data (multi-dimensional patterns) is properly estimated. A maximum-likelihood criterion with respect to the number of kernels is not useful because it tends to use a kernel to describe each pattern.

The so called model-selection problem has been addressed in many ways. Some approaches start with a few number of kernels and add new kernels when necessary. For instance, in [22], the kurtosis is used as a measure of non-Gaussianity, yielding a test for splitting a kernel in one-dimensional data. In [24] this method is extended to the multi-dimensional case, given the definition of multi-dimensional kurtosis proposed in [11] . This approach has same drawbacks, because kurtosis can be very sensitive to outliers. In [23] a greedy method is proposed, which performs a global search in combination with another local search whenever a new kernel is added.

Other model-selection methods start with a high number of kernels and proceed to fuse them. In [8][7], the EM algorithm is initialized with many kernels randomly placed and then the Minimum-description length principle [15]

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 149–158, 2006.

is applied to iteratively remove some of the kernels until the optimal number of them is found. In [19], the proposed algorithm is allowed both to split and merge kernels. Kernel fusion arises when many patterns have the same posterior probability of belonging to two different kernels. On the other hand, splitting is driven by the Kullback-Leibler divergence between a component density and empirical density in the neighborhood of the component. In this approach, the number of components remains unchanged. A heuristic criterion is used for new kernels initialization. In [26] two matrix decompositions based methods are proposed for initialization after split and merge operations, but the algorithm is still unable to determine the number of components. Recently, in [1] a new method is proposed when novel data points arrive one-by-one.

Another different approach to fit the parameters of the model is to develop a reversible jump Markov chain Monte Carlo (RJMCMC) algorithm. In [14] Richardson and Green used split and merge operations together with birth and death operations. The proposed methodology deals with parameter estimation and model order selection in a single paradigm in one-dimensional problems. In an image segmentation context and others high-dimensional problems, many free parameters must be estimated. So data-driven techniques with some previous clustering methods learning from data, have been proposed [3][17]. In [18] mixtures of Gaussians are used to model textured colors.

In this paper we propose a method that starting with only one kernel, finds the maximum-likelihood solution. Then it tests whether the underlying pdf of each kernel is Gaussian. Otherwise it replaces that kernel with two kernels adequately separated from each other. In order to detect non-Gaussianity we compare the entropy of the underlying pdf with the theoretical entropy of a Gaussian. After the kernel with worse degree of gaussianity has been splited in two, new EM steps are performed in order to obtain a new maximum-likelihood solution.

The paper is organized as follows: In section 2, we review the basic principles of Gaussian-mixture models and usual EM algorithm. In section 3, we present our maximum-entropy based criterion approach and the modified EM algorithm that exploits it. In section 4, we present the application of the method to the color image segmentation problem. Finally, in section 5 we present our conclusions and future work.

2 Gaussian-Mixture Models

2.1 Definition

A d-dimensional random variable $\mathbf{y}$ follows a finite-mixture distribution when its pdf $p(\mathbf{y}|\Theta)$ can be described by a weighted sum of known pdf's named kernels. When all of these kernels are Gaussian, the mixture is named in the same way:

$$p(\mathbf{y}|\Theta) = \sum_{i=1}^{K} \pi_i p(\mathbf{y}|\Theta_i) \tag{1}$$

where $0 \leq \pi_i \leq 1, i = 1, ..., K$, and $\sum_{i=1}^{K} \pi_i = 1$, being K the number of kernels, $\pi_1, ..., \pi_k$ the *a priori* probabilities of each kernel, and Θ_i the parameters

describing the kernel. In Gaussian mixtures, $\Theta_i = \{\mu_i, \Sigma_i\}$, that is, the average vector and the covariance matrix.

The set of parameters of a given mixture is $\Theta \equiv \{\Theta_1, ..., \Theta_k, \pi_1, ..., \pi_k\}$. Obtaining the optimal set of parameters Θ^* is usually posed in terms of maximizing the log-likelihood of the pdf to be estimated:

$$\ell(Y|\Theta) = \log p(Y|\Theta) = \log \prod_{n=1}^{N} p(y_n|\Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(y_k|\Theta_k). \quad (2)$$

With $\Theta^* = \arg\max_\Theta \ell(\Theta)$ and $Y = \{y_1, ...y_N\}$ is a set of N i.i.d. samples of the variable Y.

2.2 EM Algorithm

The EM (Expectation-Maximization) algorithm [6] is an iterative procedure that finds maximum-likelihood solutions to problems containing *hidden variables*. In the case of Gaussian mixtures [13], these variables are a set of N labels $Z = \{z^1, ..., z^N\}$ associated to the samples. Each label is a binary vector $z^i = [z_1^{(i)}, ..., z_k^{(i)}]$, with $z_m^{(i)} = 1$ and $z_p^{(i)} = 0$ if $p \neq m$, indicating that $y^{(i)}$ has been generated by the kernel m. The EM algorithm generates a sequence of estimations of the set of parameters $\{\Theta^*(t), t = 1, 2, ...\}$ by alternating two steps (Expectation and Maximization) until convergence.

The Expectation step consists of estimating the expected value of the hidden variables given the visible data Y and the current estimation of the parameters $\Theta^*(t)$. The probability of generating $\mathbf{y}_n$ with the kernel k is given by

$$p(k|\mathbf{y}_n) = \frac{\pi_k p(\mathbf{y}^{(n)}|k)}{\Sigma_{j=1}^{K} \pi_j p(\mathbf{y}^{(n)}|k)} \quad (3)$$

In the Maximization step, the new parameters $\Theta^*(t+1)$ are re-estimated, given the expected Z:

$$\begin{aligned} \pi_k = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} p(k|\mathbf{y}_n), \mu_k = \frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n)\mathbf{y}_n}{\sum_{n=1}^{N} p(k|\mathbf{y}_n)}, \\ \Sigma_k = \frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n)(\mathbf{y}_n - \mu_k)(\mathbf{y}_n - \mu_k)^T}{\sum_{n=1}^{N} p(k|\mathbf{y}_n)}, \end{aligned} \quad (4)$$

A detailed description of this classic algorithm is given in [13]. Here we focus on the fact that if K is unknown beforehand, it cannot be estimated through maximizing the log-likelihood because $\ell(\Theta)$ grows with K. In a classical EM algorithm, with a fixed number of kernels, density can be underestimated giving a poor description of the data. In the next section we describe the use of entropy to test whether a given kernel describes properly the underlying data.

3 Entropy Based EM

Entropy is a basic concept in information theory. The entropy of a given variable Y can be interpreted in terms of information, randomness, dispersion, and so on [4][16]. For a discrete variable we have:

$$H(Y) = -E_y[\log(P(Y))] = -\sum_{i=1}^{N} P(Y = y_i) \log p(Y = y_i). \quad (5)$$

where $y_1, ..., y_N$ is the set of values of a random variable Y. A fundamental result of information theory is that Gaussian variables have the maximum entropy among all the variables with equal variance. Consequently, the entropy of the underlying distribution of a kernel should reach a maximum when such a distribution is Gaussian. This theoretical maximum entropy is given by:

$$H_{max}(Y) = \frac{1}{2} \log[(2\pi e)^d |\Sigma|]. \quad (6)$$

Then, in order to decide whether a given kernel is truly Gaussian or must be replaced by two other kernels, we compare the estimated entropy of the underlying data with the entropy of a Gaussian. The estimation of the Shannon entropy of a probability density given a set of samples has been studied widely in the past (see [2] for a detailed revision). As a summary, these methods are classified as "plug-in" if they replace a nonparametric estimation of the probability density in the entropy expression and "non plug-in", which perform a "by-pass" of the density, considering the entropy from a set of samples of the same one. In this article we make an estimation of the first type [21][20].

3.1 Entropy Estimation with Parzen's Windows

The Parzen's windows approach [12] is a non-parametric method for estimating pdf's for a finite set of patterns. The general form of these pdf's using a Gaussian kernel and assuming diagonal covariance matrix $\psi = Diag(\sigma_1^2, ...\sigma_{N_a}^2)$ is:

$$P^*(Y, a) \equiv \frac{1}{N_a} \sum_{y_a \in a} K_\psi(y_b - y_a), \quad (7)$$

where

$$K_\psi(y_b - y_a) = \frac{1}{\prod_{i=1}^{d} \sigma_i (2\pi)^{\frac{d}{2}}} \prod_{j=1}^{d} \exp\left\{ -\frac{1}{2} \left(\frac{y^j - y_a^j}{\sigma_j} \right)^2 \right\}, \quad (8)$$

and a is a sample of the variable Y, N_a is the size of the sample, y^j represents the j-th component of y and y_a^j represents the j-th component of kernel y_a. In [21] a method for adjusting the widths of the kernels using maximum likelihood is proposed . Given the definition of entropy in Eq. 5, we have:

$$H_b(Y) \equiv -E_b[\log(P(Y))] = -\frac{1}{N_b} \sum_{y_b \in b} \log(P(y_b)) \quad (9)$$

where b is a sample of the variable Y and N_b is the size of the sample. If expression in Eq. 7 is plugged into Equation 9, then the entropy is estimated by:

$$H^*(Y) = \frac{1}{N_b} \sum_{y_b \in b} \log\left(\frac{1}{N_a} \sum_{y_a \in a} K_\psi(y_b - y_a)) \right) \quad (10)$$

ENTROPY BASED EM ALGORITHM
Initialization: Start with a unique kernel.
$K \leftarrow 1$. $\Theta_1 \leftarrow \{\mu_1, \Sigma_1\}$ with $\mu_1 =$ data average and $\Sigma_1 =$ data covariance.
repeat: *//Main loop*
  repeat: *//E, M Steps*
    Estimate log-likelihood in iteration i: ℓ_i
  until: $|\ell_i - \ell_{i-1}| <$ CONVERGENCE_TH
  Evaluate $H(Y)$ and $H_{max}(Y)$ globally
  if $(H(Y)/H_{max} <$ ENTROPY_TH)
    Select kernel K_* with the lowest ratio and decompose into K_1 and K_2
    Initialize parameters Θ_1 and Θ_2(Eq.13)
      Initialize new averages: μ_1 and μ_2
      Initialize new eigenvalues and eigenvector matrices: Λ_1, Λ_2, V_1 and V_2
      Set new priors: π_1 and π_2
  else Final $\leftarrow$ True
until: Final = True

Fig. 1. Our Entropy-based EM algorithm

3.2 Proposed Method

Comparing the estimations given for Eqs. 6 and 10, we have a way of quantifying the degree of Gaussianity of a given kernel. Given a set of kernels for the mixture (initially one kernel), we evaluate the real global entropy $H(y)$ and the theoretical maximum entropy $H_{max}(y)$ of the mixture by considering the individual pairs of entropies for each kernel, and the prior probabilities:

$$H(Y) = \sum_{k=1}^{K} \pi_k H_k(Y) \quad \text{and} \quad H_{max}(Y) = \sum_{k=1}^{K} \pi_k H_{max_k}(Y)\,. \tag{11}$$

If the ratio $H(y)/H_{max}(y)$ is above a given threshold we consider that all kernels are well fitted. Otherwise, we select the kernel with the lowest individual ratio and it is replaced by two other kernels that are conveniently placed and initialized. Then, a new EM with $K+1$ kernels starts.

3.3 Introducing a New Kernel

A low $H(y)/H_{max}(y)$ local ratio indicates that multi-modality arises, so the kernel must be replaced by two other kernels. The difficulty in the split step arises when the original covariance matrix needs to generate two new matrices with two restrictions: overall dispersion must remain almost constant and the new matrices must be positive definite. In [14] the split is based on the constraints of preserving the first two moments before and after the operation in one-dimensional settings. This is an ill-posed problem, because the number of equations is less than the number of unknowns and the requirement of retaining the positive definiteness of covariance matrices.

3.4 Split Equations

From definition of mixture in Eq. 1, considering that the $K*$ component is the one with lowest Gaussianity threshold, it must be decomposed into the K_1 and K_2 components with parameters $\Theta_{k_1} = (\mu_{k_1}, \Sigma_{k_1})$ and $\Theta_{k_2} = (\mu_{k_2}, \Sigma_{k_2})$. The corresponding priors, the mean vectors and the covariance matrices should satisfy the following split equations:

$$\begin{gathered} \pi_* = \pi_1 + \pi_2 \\ \pi_* \mu_* = \pi_1 \mu_1 + \pi_2 \mu_2 \\ \pi_* (\Sigma_* + \mu_* \mu_*^T) = \pi_1 (\Sigma_1 + \mu_1 \mu_1^T) + \pi_2 (\Sigma_2 + \mu_2 \mu_2^T) \end{gathered} \tag{12}$$

In [25] a RJMCMC based algorithm is presented. To solve the equations showed above, a spectral decomposition of the actual covariance matrix is used and the original problem is replaced by estimating the new eigenvalues and eigenvectors of new covariance matrices from the original one. The main limitation is that all covariance matrices of the mixture are restricted to share the same eigenvector matrix. More recently, in [5]new eigenvectors matrices are obtained applying a rotation matrix.

Let $\sum_* = V_* \Lambda_* V_*^T$ be the spectral decomposition of the covariance matrix $\sum_*$, with $\Lambda_* = diag(\lambda j*^1, ..., \lambda j*^d)$ a diagonal matrix containing the eigenvalues of $\sum_*$ with increasing order, $*$ the component with the lowest entropy ratio, π_*, π_1, π_2 the priors of both original and new components, μ_*, μ_1, μ_2 the means and $\sum_*, \sum_1, \sum_2$ the covariance matrices. Let also be D a dxd rotation matrix with columns orthonormal unit vectors. D is constructed by generating its lower triangular matrix independently from $d(d-1)/2$ different uniform $U(0,1)$ densities. The proposed split operation is given by:

$$\begin{gathered} \pi_1 = u_1 \pi_*, \;\; \pi_2 = (1 - u_1)\pi_* \\ \mu_1 = \mu_* - (\textstyle\sum_{i=1}^d u_2^i \sqrt{\lambda_*^i} V_*^i) \sqrt{\frac{\pi_2}{\pi_1}}, \quad \mu_2 = \mu_* - (\sum_{i=1}^d u_2^i \sqrt{\lambda_*^i} V_*^i) \sqrt{\frac{\pi_1}{\pi_2}} \\ \Lambda_1 = diag(u_3) diag(\iota - u_2) diag(\iota + u_2) \Lambda_* \tfrac{\pi_*}{\pi_1} \\ \Lambda_2 = diag(\iota - u_3) diag(\iota - u_2) diag(\iota + u_2) \Lambda_* \tfrac{\pi_*}{\pi_2} \\ V_1 = D V_*, \;\; V_2 = D^T V_* \end{gathered} \tag{13}$$

where, ι is a d x 1 vector with all values equal to 1, $u_1, u_2 = (u_2^1, u_2^2, ..., u_2^d)^T$ and $u_3 = (u_3^1, u_3^2, ..., u_3^d)^T$ are $2d+1$ random variables needed to construct priors, means and eigenvalues for the new component in the mixture. They are calculated as

$$\begin{gathered} u_1 \sim be(2,2), u_2^1 \sim be(1, 2d), \\ u_2^j \sim U(-1,1), u_3^1 \sim be(1,d), u_3^j \sim U(0,1) \quad \text{and} \quad j = 2, ..., d \end{gathered} \tag{14}$$

A graphical description of the splitting process in the 2-D case is showed in Fig. 2. Otherwise, a completed algorithmic description of the process is showed in Fig. 1.

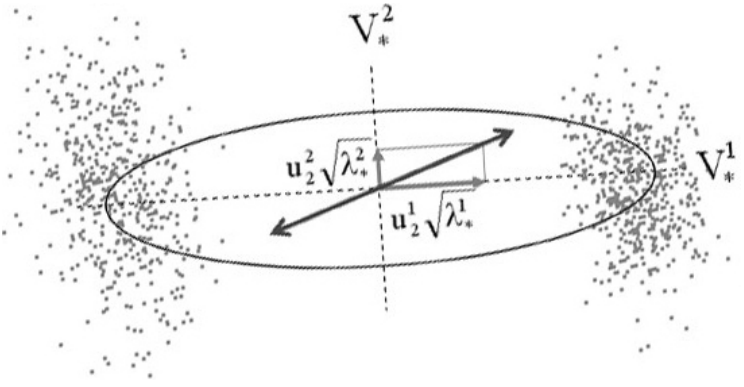


Fig. 2. 2-D Example of splitting a kernel. Directions and magnitudes of variability are defined by eigenvectors and eigenvalues of the covariance matrix.

4 Experiments

Color image segmentation is a pre-requisite to solve many computer vision problems, such as image classification, retrieval and object recognition. There are two principal approaches for image segmentation, supervised and unsupervised. The unsupervised approach consists of dividing a color image into into several homogeneous regions automatically based on some similarity measure. This is the most interesting approach. Probabilistic models like Bayesian statistics [27][9] or Markov Random Fields [10] are typically used. In [26] a split and merge EM based gaussian mixture algorithm is used but the k-means method is needed as initialization technique. Besides, the total number of components (colors in the image) is unchanged and previously established by means of a Bayesian inference criterion (BIC) because the algorithm is unable to determine the number of components.

In order to test our approach we have performed several experiments with color images. At each pixel i in the image we compute a 3-dimensional feature vector x_i with the components in the RGB color space. As a result of our algorithm we obtain the number of components (classes) M and $y_i \in [1, 2, ..., M]$ to indicate from which class the pixel i came. Therefore our image model sets that each pixel is generated by one of the gaussian densities in the gaussian mixture model. We have used different entropy thresholds and a convergence threshold of 0.1 for the EM algorithm. The sample size for estimating entropy through Parzen has been 75. We have found that despite this small size, entropy estimation is good enough. Our algorithm converges after few iterations (depending on the selected entropy threshold) finding an increasing number of kernels. In Fig. 3 we show some color image segmentation results with four different images with sizes of 189x189 pixels, resulting a set of 35.721 samples. Column one shows the original image and columns two, three and four show the resulting segmented image with increasing entropy threshold. The greater it is the demanded threshold the higher is the number of kernels generated and therefore, the number of colors detected in the image. No

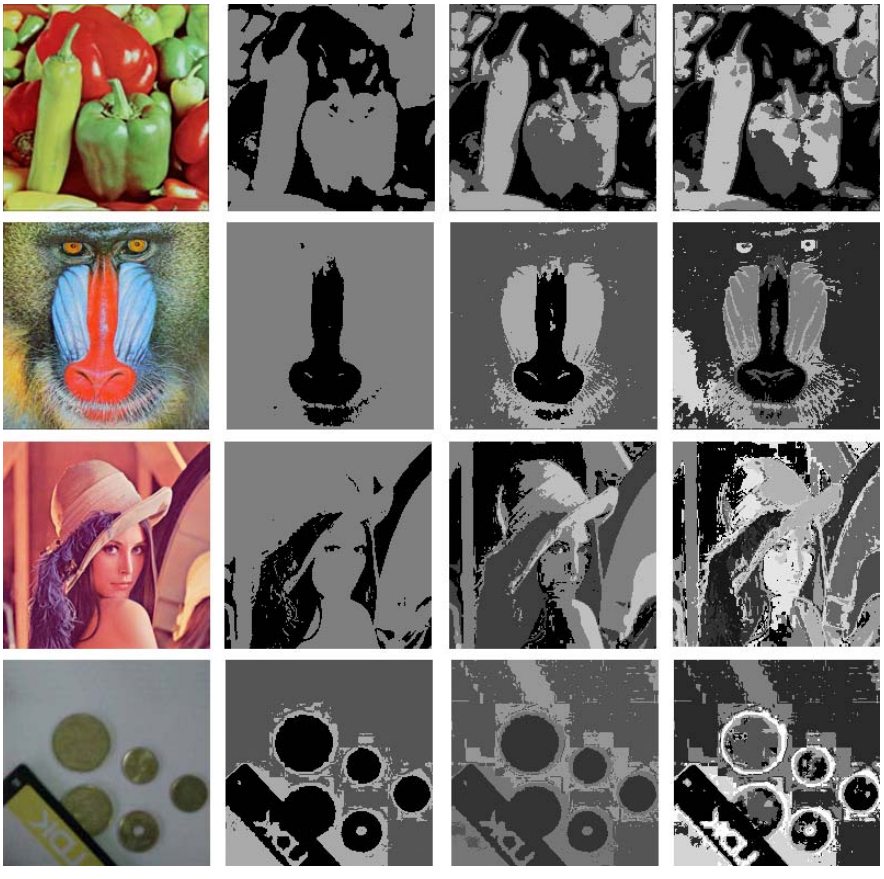


Fig. 3. Color image segmentation results. Original images (first column) and color image segmentation with different Gaussianity levels.

postprocess has been applied to the resulting images and each point was labeled with the color of the kernel to which it belonged in greater degree.

Each row shows three different color segmentation results with two classes (column two), three classes (column three) and four classes (column four) as a result of executing our algorithm with entropy thresholds of 0.65, 0.75 and 0.85 respectively.

5 Conclusions

In this paper we have presented a method for finding the optimal number of kernels in a Gaussian Mixture based on maximum entropy. We start the algorithm with only one kernel and then we decide to split it on the basis of the entropy of the underlying pdf. Our algorithm converges after a few iterations

and is suitable for unsupervised color image segmentation. The algorithm is initialized with a unique kernel whose parameters of average and covariance are given by the sample (complete image). Consequently, the algorithm is not prone to initialization. In our case, the criterion of Gaussianity is more versatile that fixing the number of kernels beforehand. For instance, one may assume that in a image sequence of the same environment, the Gaussianity threshold may be nearly constant whereas the number of kernels will be different for each frame. Although one may use the number of colors, our kernels are introduced dynamically avoiding a local minimum. We are currently addressing experiments to learn the relation between the Gaussianity threshold and the image.

References

1. O. Arandjelovic and R. Cipolla. Incremental learning of temporary-coherent gaussian mixture models. In *British Machine Vision Conference*. British Machine Vision Association (BMVA), 2005.
2. E. Beirlant, E. Dudewicz, L. Gyorfi, and E. Van der Meulen. Nonparametric entropy estimation. *International Journal on Mathematical and Statistical Sciences*, 6(1):17–39, 1996.
3. E. Clark and A. Quinn. A data-driven bayesian sampling scheme for unsupervised image segmentation, 1999. In Proceedings of ICASSP 1999.
4. T. Cover and J. Thomas. *Elements of Information Theory*. J. Wiley and Sons, 1991.
5. P. Dellaportas and I. Papageorgiou. Multivariate mixtures of normals with unknown number of components. *Statistics and Computing*, To appear.
6. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data via the em algorithm. *Journal of The Royal Statistical Society B*, 39(1):1–38, 1977.
7. M.A.T. Figueiredo and A.K. Jain. Unsupervised selection and estimation of finite mixture models, 2000. In Proceedings of the International Conference on Pattern Recognition. ICPR2000 (Barcelona).
8. M.A.T. Figueiredo, J.M.N. Leitao, and A.K. Jain. On fitting mixture models. *Hancock, E.R., Pelillo, M. (eds.): Energy Minimization Methods in Computer Vision and Pattern Recognition. Lecture Notes in Computer Science*, 1654(1):54–69, 1999.
9. J. Hornegger and H. Niemann. A novel probabilistic model for object recognition and pose estimation. *Pattern Recognition Artif. Intell.*, 15(2):241–253, 2001.
10. S.Z. Li. *Markov random field modeling in computer vision*. Springer-VerlagLondon, U.K., 1995.
11. K.V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3):519–530, 1970.
12. E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(1):1065–1076, 1962.
13. R.A. Redner and H.F. Walker. Mixture densities, maximum likelihood, and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.
14. S. Richardson and P.J. Green. On bayesian analysis of mixtures with unknown number of components (with discussion). *Journal of the Royal Statistical Society B*, 59(1):731–792, 1997.
15. J. Rissanen. Stochastic complexity in statistical inquiry, 1989. World Scientific.

16. C. Shannon. A mathematical theory of comunication. *The Bell System Technical Journal*, 27(1):379–423, 1948.
17. Z. Tu, S.C. Zhu, and H.Y. Shum. Image segmentation by data driven markov chain montecarlo, 2001. In Proceedings of ICCV 2001.
18. Z. W. Tu and S. C. Zhu. Image segmentation by datadriven markov chain monte carlo. *IEEE Trans. Pattern Anal. Mach. Intell*, 24(1):657–673, 2002.
19. N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. Smem algorithm for mixture models. *Neural Computation*, 12(1):2109–2128, 2000.
20. P. Viola, N. N. Schraudolph, and T. J. Sejnowski. Empirical entropy manipulation for real-world problems. *Adv. in Neural Infor. Proces. Systems*, 8(1), 1996.
21. P. Viola and W. M. Wells-III. Alignment by maximization of mutual information. In *5th Intern. Conf. on Computer Vision*. IEEE, 1995.
22. N. Vlassis and A. Likas. A kurtosis-based dynamic approach to gaussian mixture modeling. *IEEE Trans. Systems, Man, and Cybernetics*, 29(4):393–399, 1999.
23. N. Vlassis and A. Likas. A greedy em algorithm for gaussian mixture learning. *Neural Processing Letters*, 15(1):77–87, 2002.
24. N. Vlassis, A. Likas, and B. Krose. A multivariate kurtosis-based dynamic approach to gaussian mixture modeling, 2000. Inteligent Autonomous Systems Tech. Report.
25. Z. Zhang, K.L. Chan, Y. Wu, and C. Chen. Learning a multivariate gaussian mixture models with the reversible jump mcmc algorithm. *Statistics and Computing*, 14(1):343–355, 2004.
26. Z. Zhang, C. Chen, J. Sun, and K.L. Chan. Em algorithms for gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36(1):1973–1983, 2003.
27. S.C. Zhu, R. Zhang, and Z. Tu. Integrating bottom-up for object recognition by data driven markov chain montecarlo. In *CVPR*. IEEE, 2000.

An Image Analysis Methodology Based on Deterministic Tourist Walks

Mônica G. Campiteli[1], Alexandre S. Martinez[1], and Odemir M. Bruno[2]

[1] Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP)
Universidade de São Paulo (USP) Avenida Bandeirantes, 3900
14040-901 Ribeirão Preto SP, Brazil
monicacampiteli@pg.ffclrp.usp.br, asmartinez@ffclrp.usp.br
[2] Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP) Avenida do Trabalhador São-carlense, 400
13560-970 São Carlos SP, Brazil
bruno@icmc.usp.br

Abstract. Textures are important visual attribute used in image analysis. This paper presents a novel methodology, based on a deterministic walk, to texture analysis and texture characterization. Most of the methods adopted to classify textures deal with a defined scale of texture. The method proposed explores the set in all scales and is able to characterize efficiently different texture classes. The paper presents the deterministic walk technique and its results for two experiments using Brodatz images.

1 Introduction

Image Analysis is a field of computer vision and artificial intelligence, responsible for the extraction of meaningful information from images. The texture is an important visual attribute used in image analysis and have a broad range of applications, such as: aid of diagnoses in medical images, remote sensing, analysis of geological images and microscope images.

Although there is not a formal specification of the texture analysis, we know that this attribute is directly related to the distribution of pixels in a certain region of the image, representing an important source of information.

Textures can be defined as micro and macro textures according to the size of the set of pixels analyzed. Most of the techniques used in image retrieval are devoted to the micro textures analysis [1,2]. The number of methods applied to macro textures is still restricted, due to the inherent difficulty in the analysis [3].

In this work we present a new method for texture characterization based on a deterministic walk. This method is able to explore the image in all scales simultaneously.

2 Deterministic Tourist Walk

Although not as thoroughly studied as random walks [4,5], the study of deterministic walks has attained the interest of researchers [6,7,8,9] and new applications are

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 159–167, 2006.

being considered [10]. Here, we are interested in exploring a partially self-avoiding deterministic walk algorithm, known as the tourist walk (TW) [11,12,13,14,15] for image analysis purposes.

The tourist walk algorithm can be pictorially viewed as a tourist wishing to visit N cities randomly distributed in a map of d dimensions. The tourist starts his route in a given city of this map and moves according to the following deterministic rule: *go to the nearest city, which has not been visited in the last μ time steps*. For $\mu \geq 1$, self-avoidance is limited to the memory window $\tau = \mu - 1$, which represents a characteristic time to the city to become attractive to the tourist again (refractory time). The trajectories can intersect outside this memory range. The tourist trajectory consists of a transient part of length t (new cities are visited) and a final cycle of period p (no new cities are visited any longer). The trajectory is complex and depends strictly on the configuration of the data set. The only relation that holds is $p \geq \mu + 1$. The tourist movements are entirely performed based on a neighborhood table. This table neglects the distances among the cities, it only considers the neighborhood rank. This feature leads to invariance in scale transformations.

We call the attention to several aspects of these deterministic walks:

1. At each time interval the tourist moves from one city to another, regardless the distance between them.
2. Starting from different cities in the map, the tourist performs different trajectories with variable transient times (which can even be null $t = 0$) and ends in cycles with period $p \geq \mu + 1$.
3. The trajectories are different for different initial conditions.

Although easy to formulate and to implement numerically, according to the chosen memory window μ, this algorithm may present a complex behavior.

This complex behavior can be captured with the transient time and period joint distribution $S_{d,\mu}^{(N)}(t,p)$. Here we show that the joint distribution $S_{d,\mu}^{(N)}(t,p)$ can be efficiently used as features for image analysis.

The deterministic tourist walk with memory $\mu = 0$ is trivial since the walker does not move at each time step. The joint distribution is simply given by:

$$S_{d,0}^{(N)}(t,p) = \delta_{t,0}\delta_{p,1} \ , \tag{1}$$

where $\delta_{i,j}$ is the Kronecker's delta.

With memory $\mu = 1$, the walker must leave the visited city at each time step. The transient and period joint distribution is obtained for $N \gg 1$ [16]:

$$S_{d,1}^{(\infty)}(t,p) = \frac{\Gamma(1 + I_d^{-1})(t + I_d^{-1})}{\Gamma(t + p + I_d^{-1})} \ \delta_{p,2} \ , \tag{2}$$

where $t = 0, 1, 2, \ldots$, $\Gamma(z)$ is the gamma function and $I_d = I_{1/4}(1/2, (d+1)/2)$ is the normalized incomplete beta function. This memoryless rule ($\mu = 1$) does not lead to exploration of the random medium since after a short transient, the tourist gets trapped in pairs of cities that are mutually nearest neighbors.

Interesting phenomena occur when greater memory values are considered. In this case, the cycle distribution is no longer peaked at $p_{min} = \mu + 1$, but presents a whole spectrum of cycles with period $p \geq p_{min}$, with possible power-law decay [11,12,13].

Determinism imposes serious restrictions as it can be seen in $\mu = 2$ one dimensional systems, where all odd periods above $p_{min} = 3$ are forbidden, as well as the period $p = 6$. This study has also been carried when the deterministic rule is relaxed [17,18].

3 Tourist Walk in NonRandom Landscapes

In the context of images ($d = 2$), one can consider each pixel as a point (or city). In the original algorithm the neighborhood table has size of the order N^2, where each point interacts with all other $N - 1$ points.

For images, the algorithm has been modified since each pixel interacts only with its first and second neighbors and the walker goes always to the direction of minimum intensity difference (gradient). Open boundary condition have been employed so that each pixel has at most 8 neighbors. Thus the neighborhood table has maximum size $N \times 8$.

Starting from each pixel the algorithm computes the transient part and detects the cycle period of the trajectory building up the joint distribution. Figure 1 depicts an example of an incomplete walk in an image. The occurrence of ties is resolved by choosing the first pixel in the counter clockwise direction, preserving the deterministic nature of the algorithm.

Notice that different from the original problem, the tourist walk is not performed in a random media but in a non-disordered medium, the image. Similarly to the previous studies, the image analysis consists exploring the properties of the joint distribution (see Figure 4) over the image and compose a texture signature curve.

4 Experiment

The TW method has been applied to images from the book of Brodatz [19], a set largely used in computer vision and image processing literature as benchmark for texture analysis. Using Brodatz's images, we have performed two experiments. Firstly, we have used six classes, each one with 4 images samples. This experiment, with a low number of classes, have been performed to illustrate some important properties of the tourist deterministic walks applied to texture analysis. In a second experiment, we have used 37 classes with 9 samples each. This experiment has been idealized to show the high potential of the method to analyze and characterize texture images. Each image used on both experiments has 200x200 pixels with 256 gray levels. Figure 2 shows the classes of Brodatz's images used on the first set of experiments.

The experimental results and the discussion about the performance of the tourist deterministic walks are presented in the following.

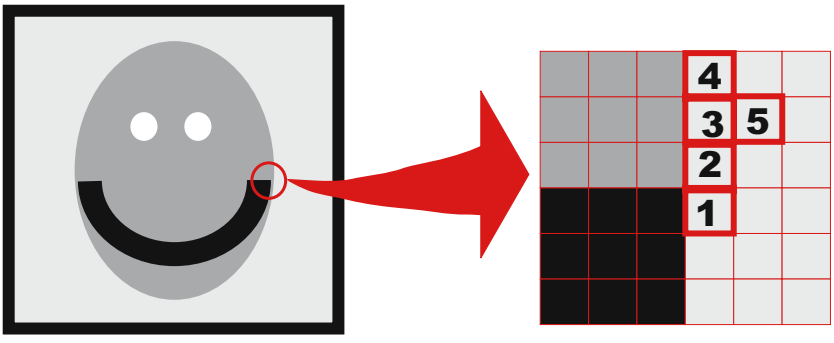


Fig. 1. Starting from pixel 1 the tourist moves to the pixel with the minimal intensity difference belonging to its first or second neighborhood. In this example, there are 6 visiting possibilities in the first moving. The tourist moves to the first pixel in the counter clockwise direction - pixel 2.

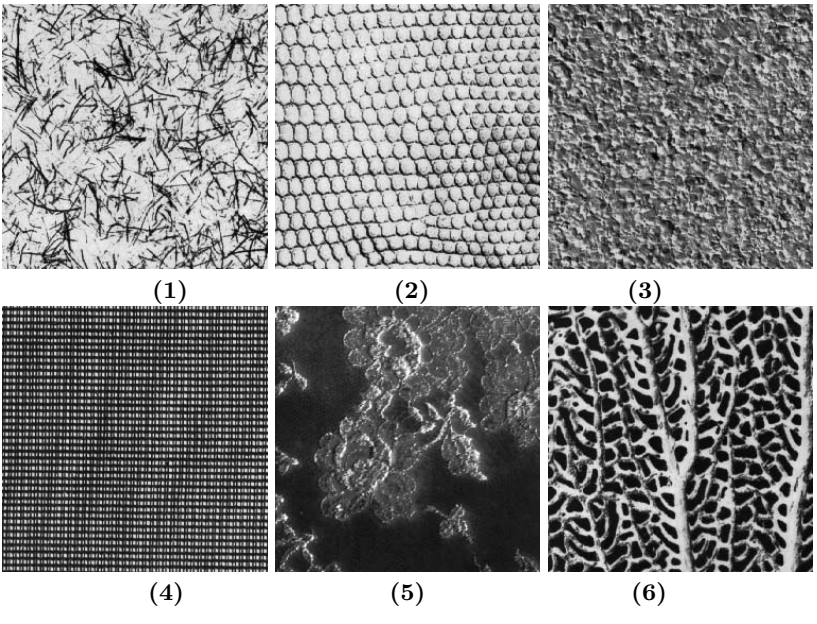


Fig. 2. Six Brodatz's classes of experiment 1

5 Results

In both experiments the Brodatz's images have been characterized by the TW texture signature. Based on these features it has been performed a data analysis by a discriminant analysis technique. Statistical analysis was made with the R 2.1.1 system [20] and the employed technique has been the *flexible discriminant analysis* (FDA) [21].

In the first set of experiments 6 Brodatz's images classes have been employed (see Figure 2).

The results of the transient time distribution ($S^{(N)}_{d,\mu}(p) = \sum_p S^{(N)}_{d,\mu}(t,p)$) are shown in Figure 3 for the texture 3. One can see that the transient time distribution is strongly dependent on the value of μ for small ($\mu < 10$) and that it approaches an assimptotic curve for μ above 10. Although these distributions are dependent on the memory, Figure 3 shows that the variation occurs for all the classes of textures (data shown for only two classes) meaning that it is possible to distinguish between them despite of the parameter value chosen.

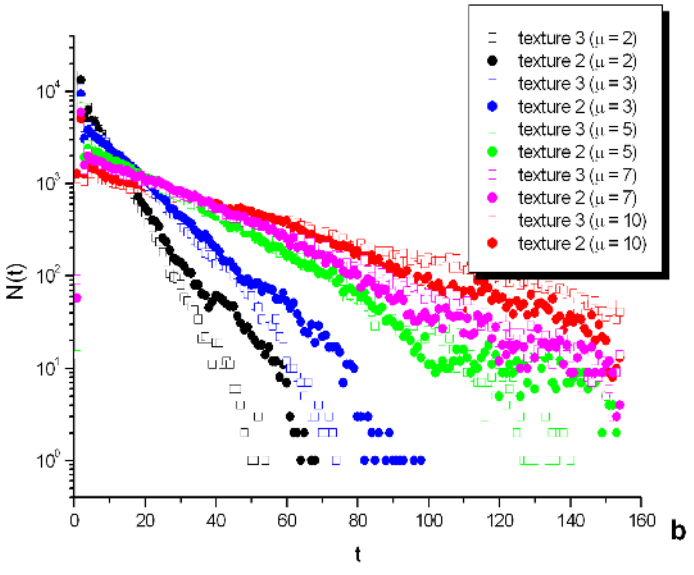


Fig. 3. Distribution of transient times for textures 2 and 3 as a function of the memory μ. The resulting curves are the average over 4 images.

Figure 4 shows six plots of the joint distribution for the TW with $\mu = 7$. Each panel corresponds to an image class used in the experiment. One can observe that the behavior of the curves are strongly different for each class of images. This illustrates the great potential of the TW technique to provide information to image analysis. The signature curves obtained from the joint distributions have been used to characterize and classify the images used in the experiment.

The FDA has been performed as follows: a feature vector has been constructed for each image sample using the first and the second transient time for each period p, with $p \geq \mu + 1$. A m-period concatenated vector has been created as:

$$\boldsymbol{S}_{2,\mu}(m) = [S^{(N)}_{2,\mu}(0:1,\mu+1)S^{(N)}_{2,\mu}(0:1,\mu+2)\ldots S^{(N)}_{2,\mu}(0:1,\mu+m)]\,. \qquad (3)$$

FDA is applied over these feature vectors. The results for the first experiment with two discriminants are show in Figure 5 as a function of μ. For the four pa-

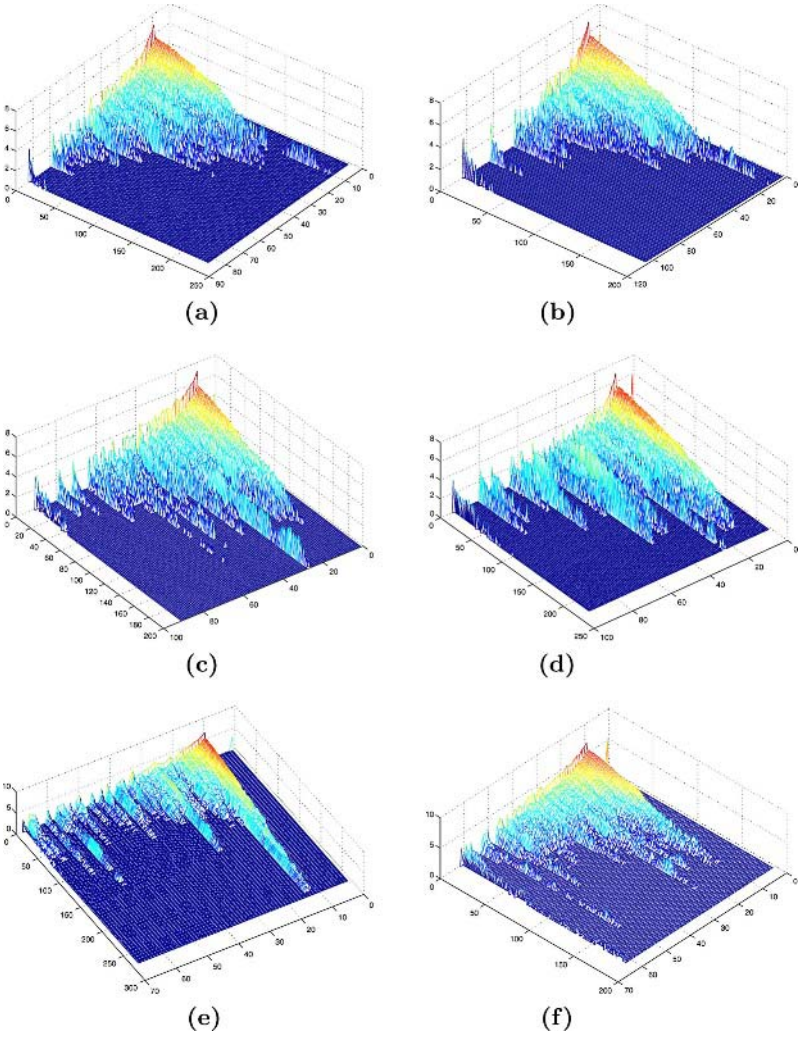


Fig. 4. Instances of joint distributions for Brodatz's classes. Panels from (a) to (f) correspond to the classes 1 to 6, respectively.

rameter values used in this experiment the classification rate of success is 100%, and the clusters have been satisfactorialy separated with only two discriminants.

Besides classifying the patterns correctly for different μ values, the results demonstrate another important property of TW. Using different μ values, one observes that the patterns of the discriminants cluster are completely different. This indicates that for different μ values, the TW extracts different information from the image. This TW property can be used to improve its classification capacity. It can also be used to compose larger feature vectors formed by the concatenation of signature curves from TWs with different memories μ. In this

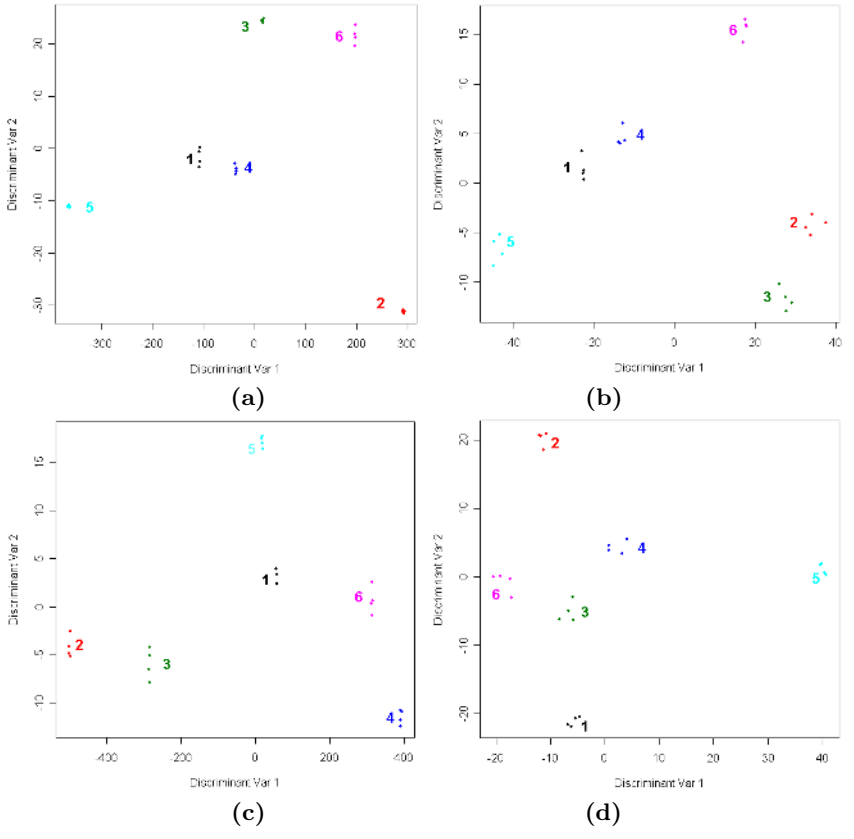


Fig. 5. Graphs for plots of the first two discriminants for different memory values: (a) $\mu = 2$, (b) $\mu = 3$, (c)$\mu = 7$ and (d) $\mu = 10$

way, tourist walks can be used to extract several vectors of characteristics, each one related to a memory, and these vectors can be combined to result in a better characterization or still be used with the intention of refining an initial classification in an iterative framework.

The second set of experiments have been performed to investigate the TW under a large number of texture classes. Similarly to the first experiment, using the TW texture signature curve with $\mu = 2$ for the feature vector, the discriminant analysis have been performed.

The results of the analysis are presented on the Table 1, where 18 discriminants have been used to accomplish the classification and the TW method reached a tax of success of 90.4%. However it is important to observe that the success tax could be better if feature vectors have been combined with different μ values or still with other pattern recognition methodology, such as for instance artificial neural network [22].

Table 1. Confusion matrix showing the classification results for 37 classes of texture images. Classification error = 0.096.

true object	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
01	**6**												1																									
02		**7**							1																													
03			**9**																																		1
04		1		**5**									1																									
05					**9**																																
06						**9**																															
07							**9**																									1					
08				2				**7**																													
09									**9**																												
10										**9**																					1						
11											**9**																										
12												**9**																			1						
13													**5**																								
14														**9**																							
15													1		**7**																						
16																**9**																					
17													1				**9**																				
18																		**9**																		
19																			**9**																	
20																				**9**																
21		1																			**8**	1														
22																					**7**	1		1												
23																						**8**														
24																							**7**							2						
25																								**9**												
26																									**9**											
27																										**8**										
28																											**8**									
29																												**7**								
30																								2					**6**							
31																														**9**	1					
32																															**8**					
33																																**9**				
34																																	**9**			
35																					1	1												**9**		
36																																			**8**	
37		3		2				1					1											1												**9**

6 Conclusion

We have presented a new method of feature extraction of image textures based on the deterministic tourist walk. The methods most commonly used deals with defined scales of pixel distributions. The distribution of transient times and periods of a set of data (image) present a wide range, capturing details on the organization of pixels from the micro to the macro scales and the resulting curve is strictly related to the configuration of the data set. We have showed that the joint distribution of the TW is an efficient tool for texture classification. We have realized two experiments using the TW and discriminant analysis to classify Brodatz textures. The results presented in this paper, show the great potential of the TW to be used as a texture analysis methodology.

Acknowledgements

The authors thank Pablo D. Batista for discussions and support in the computational implementation of the algorithm. M.G.C. acknowledges support from CAPES. A.S.M. acknowledges support from CNPq (305527/2004-5) and FAPESP (2005/02408-0). O.M.B. acknowledges support from CNPq (303746/2004-1).

References

1. Li, S.: On discontinuity-adaptive smoothness priors in computer vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(6) (1995) 576–586
2. Howarth, P., Ruger, S.: Evaluation of texture features for content-based image retrieval. Lecture Notes in Computer Science **3115** (2004) 326–334
3. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using gabor filters. Pattern Recognition **24**(12) (1991) 1167–1186
4. Scholl, J., Schol-Paschinger, E.: Classification by restricted random walks. Pattern Recognition **36** (2003) 1279–1290
5. Derrida, B.: From random walks to spin glasses. Phys. D **107**(2–4) (1997) 186–198
6. Freund, H., Grassberger, P.: The Red Queens walk. Physica A **190**(3-4) (1992) 218–237
7. Bunimovich, L.A., Troubetzkoy, S.E.: Recurrence properties of Lorentz lattice gas cellular automata. J. Stat. Phys. **67**(1-2) (1992) 289–302
8. Bunimovich, L.A.: Deterministic walks in random environments. Physica D **187** (2004) 20–29
9. Gale, D., Propp, J., Sutherland, S., Troubetzkoy, S.: Further travels with my ant. Math. Inteligencer **17**(3) (1995) 48–56
10. aes Campiteli, M.G., Batista, P.D., Kinouchi, O., Martinez, A.S.: Deterministic walks as an algorithm of pattern recognition. Phys. Rev. E (submitted) (2006)
11. Lima, G.F., Martinez, A.S., Kinouchi, O.: Deterministic walks in random media. Phys. Rev. Lett. **87**(1) (2001) 010603
12. Stanley, H.E., Buldyrev, S.V.: Statistical physics - the salesman and the tourist. Nature (London) **413**(6854) (2001) 373–374
13. Kinouchi, O., Martinez, A.S., Lima, G.F., Loureno, G.M., Risau-Gusman, S.: Deterministic walks in random networks: an application to thesaurus graphs. Physica A **315**(3/4) (2002) 665–676
14. Boyer, D., Miramontes, O., Ramos-Fernandez, G., Mateos, J.L., Cocho, G.: Modeling the searching behavior of social monkeys. Physica A **342**(1-2) (2004) 329–335
15. Boyer, D., Larralde, H.: Looking for the right thing at the right place: Phase transition in an agent model with heterogeneous spatial resources. Complexity **10**(3) (2005) 52–55
16. Terçariol, C.A.S., Martinez, A.S.: Analytical results for the statistical distribution related to memoryless deterministic tourist walk: Dimensionality effect and mean field models. Phys. Rev. E **72** (2005) 021103
17. Risau-Gusman, S., Martinez, A.S., Kinouchi, O.: Escaping from cycles through a glass transition. Phys. Rev. E **68** (2003) Art. No. 016104
18. Martinez, A.S., Kinouchi, O., Risau-Gusman, S.: Exploratory behavior, trap models and glass transitions. Phys. Rev. E **69** (2004) Art. No. 017101
19. Brodatz, P.: A Photographic Album for Arts and Design. Volume 1. Dover Publishing Co., Toronto (1966)
20. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2005)
21. Webb, A.: Statistical Pattern Recognition. Volume 1. Newnes (1999)
22. Duda, R.O., Hart, P.E.: Pattern Classification. John Wiley and Sons (2000)

Feature Characterization in Iris Recognition with Stochastic Autoregressive Models

Luis E. Garza Castañón[1], Saúl Montes de Oca[2], and Rubén Morales-Menéndez[1]

[1] Department of Mechatronics and Automation, ITESM Monterrey Campus
{legarza, rmm}@itesm.mx
[2] Automation Graduate Program Student, ITESM Monterrey Campus
Av. Eugenio Garza Sada Sur No. 2501
Monterrey, N.L. 64849 México
saul_montesdeoca@yahoo.com.mx

Abstract. Iris recognition is a reliable technique for identification of people. A typical iris recognition system is composed of four phases: image acquisition and preprocessing, iris localization and extraction, iris features characterization, and comparison and matching. We are introducing stochastic autoregressive with exogenous inputs models for the features characterization step. Every model is learned from data. In the comparison and matching step, data taken from iris sample are substituted into every model and residuals are generated. A decision is taken based on a threshold calculated experimentally. A successful rate of identifications for UBIRIS and MILES databases shows potential applications.

1 Introduction

The main intention of biometrics systems is to provide reliable automatic recognition of individuals based on the measuring of a physical or behavioral characteristic of persons. Biometrics can be used for access control to airports and military installations, or to access personal equipments such as laptops and cellular phones. Other applications are related to public applications, such as banking operations [12], refugee control and missing children. A wide variety of biometrics systems have been deployed and they include different human features such as: face, fingerprint, hand shape, palmprint, signature, voice and iris [9].

Iris recognition may provide the best solution by offering a much more discriminating power than the others biometrics. Specific characteristics of iris such as a data-rich structure (even the left and right iris of a single person, seems to be highly distinctive), genetic independence (while the general structure of the iris is genetically determined, the particulars of its minutiae are critically depend on circumstances highly unlikely to be replicated via the natural course of events), stability over time (following adolescence, the healthy iris varies little for the rest of a person's life) and physical protection, makes the use of iris recognition as biometric well recognized. The spatial patterns that are apparent in the human iris are highly distinctive. Also, its image is relatively insensitive to illumination, and changes in viewing angle cause only affine transformations;

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 168–177, 2006.

even the nonaffine pattern distortion generated by pupillary dilation is readily reversible. Finally, it is easy to isolate its features and size-invariant representation, [2]. Automated iris recognition is an alternative for noninvasive verification and identification of people.

The paper is organized as follows. Section 2 reviews the state-of-the-art. Section 3 presents our proposal. Section 4 shows the experimental results. Section 5 compares and discusses the results. Finally, section 6 concludes the paper.

2 State of the Art

In last years, there have been different implementations of iris recognition systems; some commercial products have been evaluated by requirement of the US department of homeland security [8]. The most successful system is the well known Daugman's system [2], that used multi-scale quadrature wavelets (Gabor filters) for extracting texture phase structure information of the iris. A $2,048$-bit iris code is generated and compared; the difference between a pair of iris representations by their Hamming distance are computed.

In [5] iris features are extracted by applying a fast procedure of iris localization and alignment with circular contours. Features extraction is achieved by application of Laplacian filtering and finally correlation coefficients for matching are computed. A method based on Sobel transformation for features extraction and classification with wavelet probabilistic neural networks is reported in [1]. In [7] a new method is presented for noise elimination in iris images; noise such as eyelashes, pupil, eyelids and reflections. The approach is based on the fusion of edge and region information. In [3] an iris recognition approach based on mutual information (MI) is developed. In that work, pair of iris samples are geometrically aligned by maximizing their MI, subsequently eyes iris are recognized.

We followed standard techniques in iris recognition such as integro-differential operators for iris localization, and histogram equalization over extracted iris area for illumination variation compensation. Our contribution is in the step of iris features characterization by using stochastic AutoRegresive with eXogenous variable (ARX) models, commonly used in the Control Engineering community. Every eye iris is represented by an ARX model, the model is learned from data. The comparison and matching step are made based on the generated residuals of the ARX models. A decision (accept or reject) is taken based on the residuals and a threshold experimentally computed. The architecture of the proposed method is shown in Fig. 1.

We tested our ideas with *UBIRIS* database [13] and *MILES* database [11]. Under certain conditions, we found a rate of successful identifications of 99.7 % and 100 % respectively.

3 The Proposed Approach

To implement our approach we rely on the use of colored eyes images from *UBIRIS* and *MILES* databases. The databases include samples where iris is free

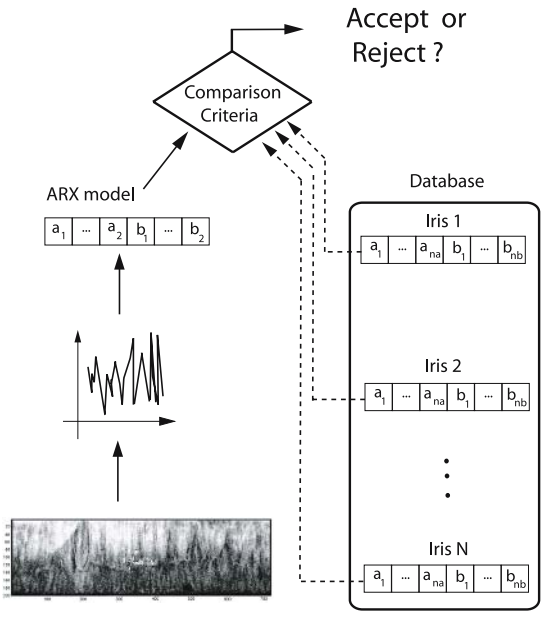


Fig. 1. Architecture of the iris recognition system based on ARX models

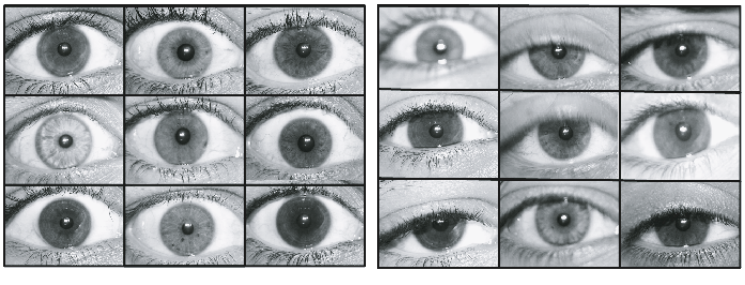


Fig. 2. Eye iris samples. Left picture shows clean samples (eye images have no noise), right picture presents noisy samples (eye images with moderate obstruction).

from any occlusion, and others with moderate obstruction because of eyelids and eyelashes, Fig. 2. We transform the colored images representation to grey level pixels (it is sufficient to reveal the relevant features of iris .)

Our iris recognition system consists, as in other typical systems, of four basic steps: iris localization and extraction, iris features characterization, comparison and matching.

Iris Localization. The search of limbic and pupilar limits is achieved with the use of the integrodifferential operator, eqn. (1), where $I(x, y)$ is an image containing an eye.

$$\max_{(r,x_0,y_0)} = \left| \frac{\partial}{\partial r} G(r) * \oint_{r,x_c,y_c} \frac{I(x,y)}{2\pi r} ds \right| \quad (1)$$

The operator behaves as an iterative circular edge detector. The operator searches over the image domain (x, y) for the maximum in the partial derivative with respect to an increasing radius r. The searching step is over the normalized contour integral of $I(x, y)$ along a circular arc ds of radius r and center coordinates (x_0, y_0). The symbol $*$ denotes convolution and $G_\sigma(r)$ is a smoothing function, $\frac{1}{\sqrt{2\pi}\sigma} e^{-(r-r_0)^2/2\sigma^2}$, typically a radial Gaussian with center r_0 and standard deviation σ. The result of this localization step is shown in Fig. 3 (left picture).

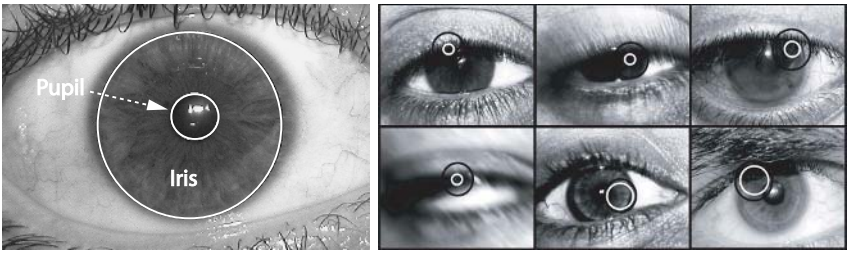


Fig. 3. Localization of limbic and pupilar limits. Left picture shows good results for a clean sample, while right picture shows failed localization for several noisy eyes images.

This operator behaves well in most cases for moderate noise conditions, but requires some fine tuning of parameters, in order to deal with pupil reflections, obscure eyes and excess of illumination. Heavy occlusion of iris by eyelashes or eyelids needs to be handled by other methods. Examples of failed localization of limbic and pupilar limits are shown in Fig. 3 (rigth picture). Eye images with heavy occlusion were discarded in this work.

The extracted iris image has to be normalized to compensate for pupil dilation and contraction under variations of illumination . This process is achieved by a transformation from cartesian to polar coordinates, using eqn. (2).

$$x(r,\theta) = (1-r)x_p(\theta) + rx_s(\theta) \quad y(r,\theta) = (1-r)y_p(\theta) + ry_s(\theta) \quad (2)$$

where $x(r, \theta)$ and $y(r, \theta)$ are defined as a linear combination of pupil limits $(x_p(\theta)$, $y_p(\theta))$ and limbic limits $(x_s(\theta)$, $y_s(\theta))$. r is defined in the interval $[0, 1]$, and θ in the interval $[0, 2\pi]$. Fig. 4 shows an example of this transformation.

Feature Characterization by ARX Models. We propose the representation of iris image by an stochastic ARX model, where AR refers to the autoregressive part and X to the extra input (called eXogeneous variable). The output v_k depends on past values and input u_k variable.

We divided the iris image strip in a rectangular grid. The mean grey level value of every subarea in the grid is represented by the output variable of the

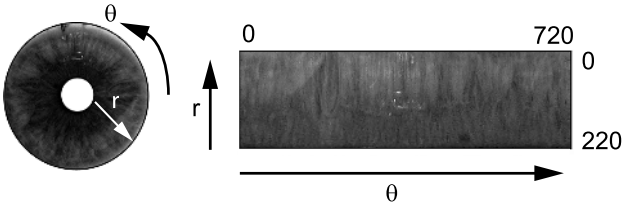


Fig. 4. Rectangular image strip

model, v_k; the corresponding row number of subarea on the grid is the input variable of the model, u_k.

A general equation of the discrete-time ARX model is given by eqn. (3).

$$v_k = a_1 v_{k-1} + \cdots + a_{n_a} v_{k-n_a} + b_1 u_{k-1-n_d} + \cdots + b_{n_b} u_{k-n_b-n_d} \tag{3}$$

where $\{a_i\}_{i=1}^{n_a}$ and $\{b_j\}_{j=1}^{n_b}$ are the model coefficients. The variable n_d is an integer number representing the number of times steps that output v_k takes to show the effect of a given input u_k; this term usually is called dead time. We assumed $n_d = 0$ for this case.

The coefficients $\{a_i\}_{i=1}^{n_a}$ and $\{b_j\}_{j=1}^{n_b}$ will be learned using the classical Least Squares (LS) algorithm. Least squares optimality has several attractive features for the purposes of identification. Large errors are heavily penalized, it can be obtained by straightforward matrix algebra, and the optimization criterion is related to statistical variance, so the properties of the solution can be analyzed according to statistical criteria. Using the following definitions:

$$\varphi_k^T = [v_{k-1}, \cdots, v_{k-na}, u_{k-1}, \cdots, u_{k-nb}] \quad \theta = [a_1, \cdots, a_{na}, b_1, \cdots, b_{nb}] \tag{4}$$

we can rewrite eqn. (3) (considering $n_d = 0$), as $v_k = \Psi_k^T \theta$. We want to find the coefficients vector θ that makes the best estimate of output v_k, $\widehat{v}_k = \varphi_k^T \theta + e_k$, where $e_k \sim \mathcal{N}(0, 1)$. Considering, $e_N = V_N - \Psi_N \theta_N$. The performance index that the LS algorithm minimizes is given by:

$$J(\theta) = \sum_{k=n_m}^{N} e_k^2 = e_N^T e_N \tag{5}$$

where $n_m = max\{n_a, n_b\}$ and N represents the number of data. The minimum value of $J(\theta)$ can be obtained when its gradient is equal to zero. Using this LS algorithm we can build a iris database, I^T, expressed in terms of ARX model for every iris.

Iris comparison. When an iris comes for recognition, an error sequence must be computed by

$$e_N = V_N^{I^T} - \Psi_N^{I^S} \theta_N^{I^T} \tag{6}$$

where $V_N^{I^T}$ and $\theta_N^{I^T}$ belongs to iris database (I^T) and $\Psi_N^{I^S}$ belongs to iris sample (I^S). After computing e_N, the comparison is made by using eqn. (5), where $\Psi_N = \left[\varphi_{n_m}^T, \cdots, \varphi_{n_{data}}^T\right]^T$.

4 Experiments

Our ideas were tested with both *UBIRIS* and *MILES* databases. Noisy images (excessive occlusion) were discarded.

Ubiris database. Using the *UBIRIS* database we selected 1,013 samples coming from 173 users. The order of ARX model (number of coefficients) was determined empirically. The best results were obtained for $n_a = 5$ and $n_b = 5$.

Design of experiments. First, we took away 10 % of the worst eyes samples (because of occlusion) and worked with the 90 % of the database; then, we took away 20 % of the worst eyes samples and worked with the 80 % of the database, and so on. For each percentage of the database, a decision threshold was experimentally learned using a number of randomly selected iris samples (learning samples). Finally, we worked with the remaining iris images as a testing samples. Fig. 5 describes graphically our design of experiments. Table 1 presents the detailed information of these experiments and Fig. 6 shows the results.

MILES database. From *MILES* experimental database we selected 213 samples from 199 users. We followed the same design of experiments shown in Fig. 5. Table 2 presents the detailed information of these experiments and Fig. 7 shows the results.

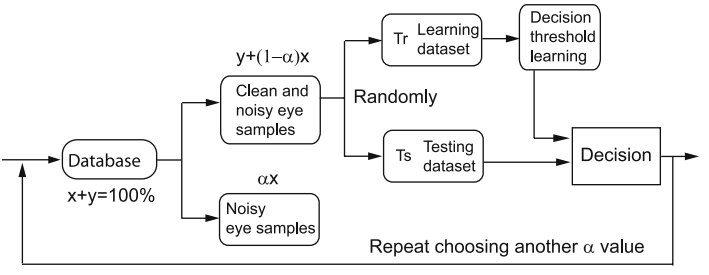


Fig. 5. Design of experiments

In order to analyze the performance of our proposal, we computed the two classical types of errors: *False Match Rate* (*FMR*) and *False non-Match Rate* (*FNMR*). Fig. 8 presents the *Receiver Operating Characteristic* (*ROC*) for *UBIRIS* experimental database, where the system performance for different thresholds are shown. Databases with cleaner iris samples shows better results.

In Fig. 9 we can see the authentic-impostor distribution curves for two *UBIRIS* databases (50, 100 %). For the impostor distribution we used pairs of samples from different persons. The genuine distribution was computed with pairs of samples from same persons. Left y-axis represents the density distribution of the genuine distribution, while right y-axis corresponds to impostor distribution.

Table 1. Experimentation with *UBIRIS* database

DB Size %	Total number eye samples to work with	Number of eye samples, Tr	Number of eye samples, Ts
100	1013	52	788
90	912	46	693
80	811	42	596
70	710	36	501
50	507	26	308

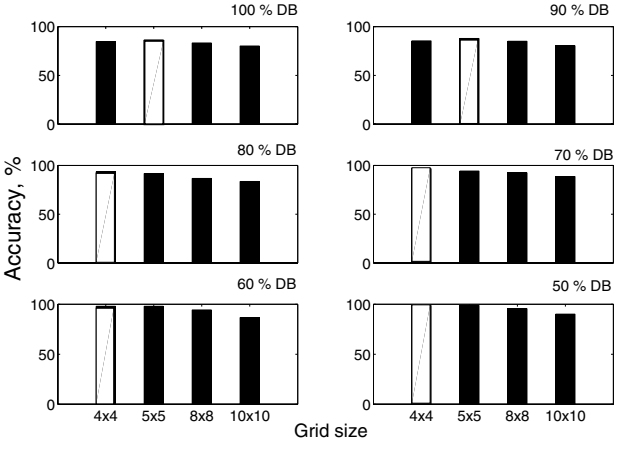


Fig. 6. Results for different percentage of *URIBIS* database. Horizontal axis represent the different grid size. For each plot a different percentage of database (DB) was used. White bar shows the best result (86.3, 87.8, 93.4, 96.7, 97.9 and 99.7 %) respectively.

Table 2. MILES Databases used for experiments with 119 users

DB Size %	Total number eye samples to work with	Number of eye samples, Tr	Number of eye samples, Ts
100	36	10	167
75	36	8	116

x-axis is the Euclidian distance. The overlapping between distribution curves in Fig. 9 (left plot) leads to worst results. Similar curves were obtained for *MILES* database experiments (not shown).

Other experiments were performed by extending the ARX models to the multivariate case. We defined two input variables, mean and variance of grey level intensity for each subarea, and two outputs, row and column location of each subarea in the iris strip. Experiments showed best results in the order of 98.9%, working with same conditions.

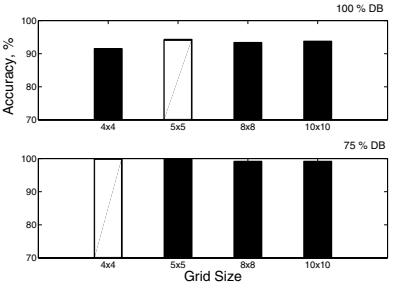


Fig. 7. Results for different percentage of *MILES* database. Horizontal axis represent the different grid size. We tested 100 % and 75 % of database (DB). White bar shows the best result (94.35 and 100 %) respectively.

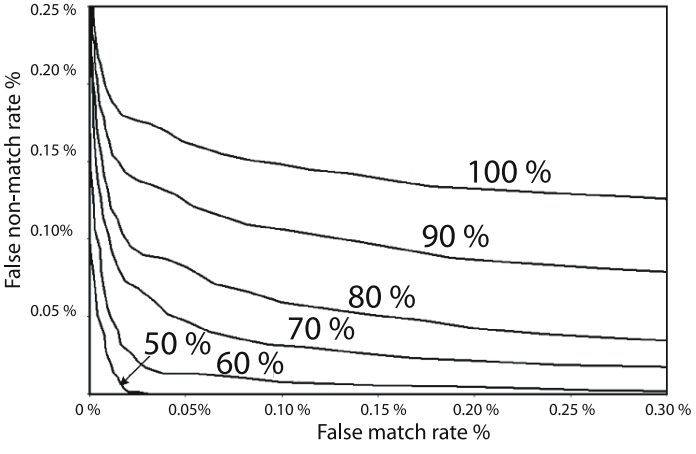


Fig. 8. *ROC* curve for experiments with different databases

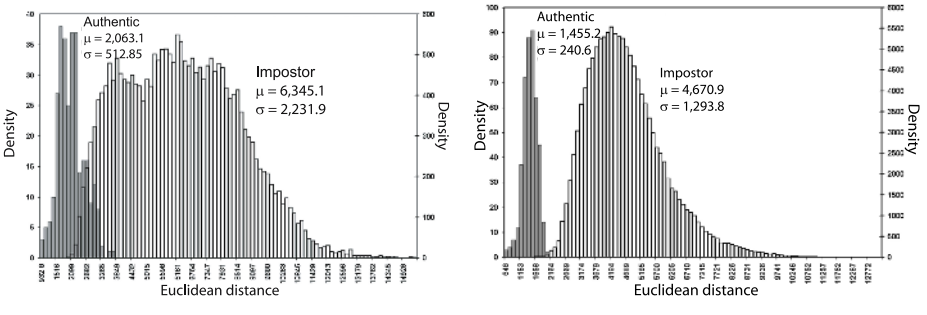


Fig. 9. Authentic-Impostor distribution for ARX based system. Left plot shows a database with 100 % of data, and right plot corresponds to 50 % of data.

5 Discussion

Daugman's system [2] has been tested thoroughly with databases containing thousands of samples, and reports of 100 % of accuracy have been given. Same order results are shown by [3], [7] and [10]; however, none of these works specify the quality of databases, so a direct comparison is not possible. We think that our proposal is competitive when clean eye images databases are used, which means eye images with no obstruction and noise. Our best results were achieved by using univariate *ARX* models and were 99.7 % for *UBIRIS* database and 100 % for *MILES* database.

6 Conclusions

A new approach for iris recognition has been presented. The novel contribution relies on the feature characterization of iris by introducing stochastic *ARX* models. Although experimental results showed better results for databases with cleaner eyes images, we are looking forward to improve the methodology by combining statistical sampling methods and stochastic models, [6].

References

1. Ch. Chen and Ch. Chu, "Low Complexity iris Recognition Based on Wavelet Probabilistic Neural Networks", In *Proc. of the Int. Joint Conf. on Neural Networks*, vol. 3, pp. 1930-1935, 2005.
2. J. Daugman, "How Iris Recognition Works", *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1), 2004, pp. 21-30.
3. M. Dobes, L. Machala, P. Tichasvky, and J. Pospisil, "Human Eye Iris Recognition Using The Mutual Information", *Optik*, No. 9, 2004, pp. 399-404.
4. G. Franklin, J. Powell, and M. Workman, Digital Control of Dynamic Systems, Addison-Wesley, 3^a edition, 1997
5. B. Ganeshan, D. Theckedath, R. Young, and C. Chatwin, "Biometric Iris Recognition System Using a Fast and Robust Iris Localization and Alignment Procedure", In *Optics and Lasers in Eng.*, Vol. 44, pp. 1-24, 2006.
6. L.E. Garza Castañon, S. Montes de Oca, and R. Morales-Menéndez "An Application of Random and Hammersley Sampling Methods to Iris Recognition", In *to appear in the 19th Int. Conf. on Industrial, Engineering & Other Applications of Applied Intelligent SystemsOptics and Lasers in Eng.*, Annecy, Fr, June 2006
7. J. Huang, Y. Wang, T. Tan, and J. Cui, "A New Iris Segmentation Method for Iris Recognition System", In *Proc. of the 17th Int Conf on Pattern Recognition*, 2004, pp. 554-557.
8. Independent Testing of Iris Recognition Technology Final Report, International Biometric Group, May 2005.
9. A. Jain, A. Ross, A. Prabhakar, "An Introduction to Biometric Recognition", *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1), 2004, pp. 4-20.
10. L. Ma, Y. Wang, T. Tan, and D. Zhang, "Personal Identification Based on Iris Texture Analysis", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(12), 2003, pp. 1519 - 1533.

11. Miles Research. Sample iris Pictures. http://www.milesresearch.com/
12. M. Negin, Chmielewski T., Salganicoff M., Camus T., Cahn U., Venetianer P., and Zhang G. "An Iris Biometric System for Public and Personal Use ", *Computer*, 33(2), 2000, pp. 70-75.
13. H. Proenca, and L. Alexandre,"UBIRIS: A Noisy Iris Image Database", in *Proc of the Int. Conf. on Image Analysis and Processing 2005*, Vol. 1, pp. 970-977.

Cryptographic Keys Generation Using FingerCodes

José A. Ramírez-Ruiz, Carlos F. Pfeiffer, and Juan A. Nolazco-Flores

Computer Science Department, ITESM, Campus Monterrey
Av. Eugenio Garza Sada 2501 Sur, Col. Tecnológico
Monterrey, N.L., México, C.P. 64849
{pfeiffer, jnolazco}@itesm.mx

Abstract. We show an architecture to automatically generate cryptographic keys using the FingerCode as defined by Jain *et al.* [4]. The FingerCode is obtained from gray scale fingerprint images. The architecture uses a classifier to compensate for the natural variability on the FingerCodes. In a training step the FingerCodes of the fingerprint samples for registered users are obtained; then random binary codes are assigned to each set of FingerCodes from the same finger, and finally an array of Support Vector Machines (SVM) is trained to associate the FingerCodes to their assigned random binary key. Each SVM is independent and assigns one bit, allowing the construction of binary keys of arbitrary length by adding and training more SVMs. To test the system, different set of fingerprint images from the same fingers used on the training step were used. The FingerCodes were calculated used as input to the SVM array to generate the assigned keys. Experimental results obtained using fingerprints selected from the FVC2000 and FVC2002 databases show results up to 90% performance on generating valid keys.

Keywords: Cryptographic Keys, Biometrics, FingerCode, Support Vector Machine.

1 Introduction

Secure applications such as electronic banking, e-commerce and access control to restricted areas, where privacy and security of information stored or transferred are vital, have become a very important topic in today's daily life. The most common way for resource access control, is the using of passwords to restrict the access to registered users and to secure critical information. The use of plain passwords is not secure enough, they have to be hashed or encrypted before being stored. Passwords can also be stolen or intercepted, or even guessed by a clever program if not carefully chosen. When secure access control is required, authentication protocols and cryptography are required. Cryptography intends to protect information by transforming the plain text into an unreadable format, called cipher text. Most modern cryptographic algorithms require the use of random binary keys for the encryption and decryption of the message. The key

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 178–187, 2006.

may not be required to be secret for the encrypting part of the algorithm (for public key algorithms), but it is always required to be secret for the decrypting part of the algorithm. For symmetric private key algorithms the key must always be kept secret. Well designed modern cryptography techniques can be virtually unbreakable.

Recently it has been proposed the use of biometrics with or instead of passwords. Biometrics is the automatic characterization of persons based on their physiological or behavioral characteristics. It is a current topic of research if they can be used as key generators in cryptosystems, where they would potentially enhance other security processes, for instance, algorithms employed for encryption, hash functions, key-stream generators, etc. Since cryptosystems based on biometrics has the advantage that the user does not have to memorize the key nor has to be stored in any place [13], then it is convenient to propose a method to generate binary keys using biometrics. From all the biometrics, fingerprint identification and authentication is the most proved, inexpensive and widely used technique [1,2]. Early work on the use of biometrics to generate binary keys is the one by Souter *et al.* [16], where they generate a key after identifying who is the user. In this method they first identify the user, and once the user is verified the cryptographic key is released. The drawback of this method is that the key information has to be stored in the system and the release of it depends only in one bit of information, making this method very susceptible to attacks. Another approach is the work by Clancy *et al.* [17], where they propose to generate the key directly from fingerprint minutiae. Minutiae, are features obtained from the terminations and bifurcations of the patterns generated by the ridges and valleys of the fingerprints. The problem with minutiae is that is very difficult to obtain a clean set, free of spurious minutiae, and to replicate the same set if the fingerprint is dirty, wet, or with any kind of moisture. Since fingerprints have quasi-parallel ridges and furrows, and well-defined local frequency and orientation that define a unique texture within each finger, Jain *et al.* [4] proposed to characterize fingerprints using FingerCodes. In their research work, they found that FingerCodes are more stable than minutiae.

The main goal of this work is to propose an architecture to assign and reproduce a random binary key based on the FingerCode. The main challenge of this research is to find suitable to repeatedly generate the same key when the same finger is applied, despite natural variations on the fingerprint acquisition process, and at the same time not to produce a given valid key when not authorized fingerprints are processed.

The architecture of the system we propose is depicted in figure 1 and it is detailed in the following sections. This architecture requires the use a classifier to compensate for the natural variations on the FingerCode coming from different sample fingerprints of the same finger. In this work we explore the use of a Support Vector Machine (SVM) as a classifier, a technique designed to classify scattered data, in which other types of classifiers find difficulties. SVM has been successfully used in other biometric tasks [12,11].

The training stage, performed off line, is shown under the dotted line. The upper part shows the on line phase (in this research it was substituted for the testing part). The biometric feature vector is generated using the FingerCode. In the training stage, a binary random number of fixed length is assigned to each user, and for each bit position a SVM is trained. Each one of the SVM is trained for all the users (each user is represented by a finger), each user providing several samples of his fingerprint. In this way, each SVM generates only a binary digit when a finger code is presented. If we train n SVMs, then we can generate n bits, which all together could be used either as the seed for a random generator or if n is large enough as a cryptographic key itself.

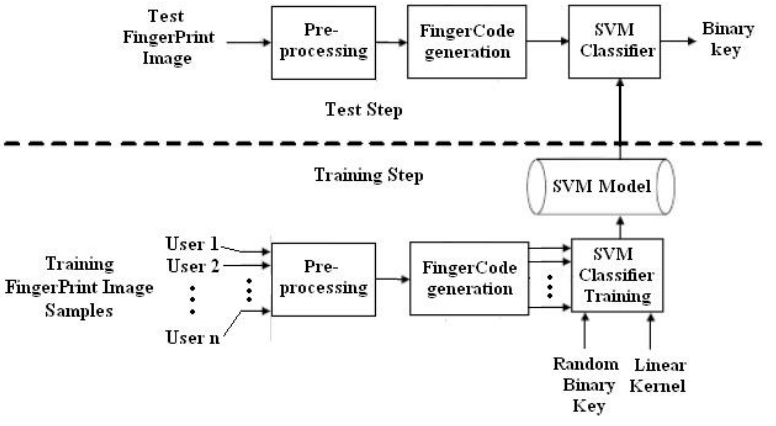


Fig. 1. System Architecture

This architecture would ideally produce the same key for the same user, compensating for the inherent variations on the FingerCodes coming from different samples of the same finger. In practice, we show that its performance varies depending on the number of users, the number of samples used for training, and the strategy used to assign the random binary codes.

In Section 2, the FingerCode calculation process is reviewed. In section 3 we briefly describe the SVM concepts. In section 4, the cryptographic fingerprint key generator architecture is explained to more detail. In section 5 we report the experimental results that show the effectiveness of the approach tested with fingerprints samples obtained from the FVC2000 and FVC2002 databases [15]. In section 6, comments and conclusions are discussed.

2 Obtaining FingerCodes

Fingerprints images are oriented textured patterns composed of ridges and valleys. User identification and verification are based on two different types of features coming from these patterns: minutiae and textures [3]. Minutiae are the terminations or bifurcations of the lines in the pattern that represents the fingerprint. On the

other hand, the fingerprint texture is characterized by the orientation, and local frequency of the ridge flow. The FingerCode is obtained by applying a bank of Gabor filters sensitive to this texture to a tessellation of the fingerprint around its core. The steps to obtain the FingerCode are as follows: given a vertically aligned gray scale fingerprint image, the first step is to find the center. The center or core of the fingerprint is defined as the point with maximum curvature change of the peaks and valley of the fingerprint. The second step, is to define a tessellation of small sectors, S_i for $i = 1, ..., n$, which are concentric to the core of the fingerprint and cover a around of it. In order to make the fingerprint images more uniform, a sector's normalization step is performed. The fingerprint image need to be normalized because when it is captured the distribution of pressure is different for each part of the fingerprint surface. This different pressure distribution causes that captured images not to be uniform. Typically normalization consists in dividing the fingerprint area in smaller region and using local histogram equalization on these regions. A better technique to normalize and enhance the fingerprint image is the using of local directional filters tuned to the mean orientation and ridge separation of each region.

The last step is to apply a bank of Gabor filters with eight different directions to each sector as explained in Jain [4]. The end step is to calculate each value of the feature vector as the average of the absolute value of the standard deviation on each filtered sector, normalized to integers from 0 to 255. These steps are shown on figure 2.

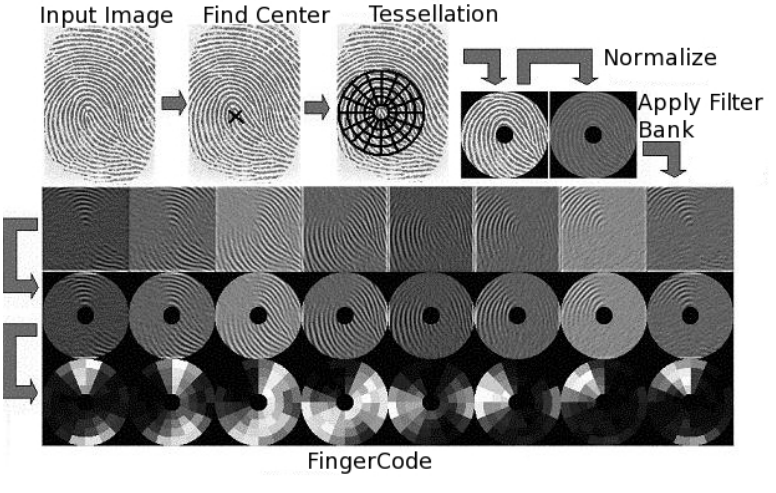


Fig. 2. Procedure to obtain the FingerCode of a fingerprint

3 Support Vector Machine

Support Vector Machine (SVM) is an algorithm used for pattern recognition. In this research it is employed as a classifier. It was first developed by Vapnik and Chervonenkis [5,7]. SVM has been used for several applications, including

biometrics [12,11]. The goal of the basic SVM is to classify input feature vectors into one of two classes. Defining the set of pairs $\{x_i, y_i\}$; where $x_i \in \mathbb{R}^n$ are the training vectors and $y_i = \{-1, 1\}$ are the labels, the SVM learning algorithm finds an hyperplane (w, b) such that,

$$\min_{x_i,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{1}$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \tag{2}$$

$$\xi_i \geq 0 \tag{3}$$

where ξ_i is a slack variable and C is a positive real constant known as a trade-off parameter between error and margin. To extend the linear method to a nonlinear technique, the input data is mapped into a higher dimensional space by function ϕ. However, exact specification of ϕ is not needed; instead, the expression known as kernel $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is defined. There are different types of kernels: linear, polynomial, radial basis function (RBF) and sigmoid. The use of non linear extensions is recommended when the data is highly inseparable or the data is not scattered in the space. In this research, we tested only SVM's using the linear kernel.

4 Proposed Models

To be able to generate a binary cryptographic key through the texture code (FingerCode) of the fingerprint we introduce 4 different models showing the advantages and disadvantages of each one. In all the models the classifier used is a set of Support Vector Machines as described previously.

4.1 Model A

In this model the existence of n users is assumed, each one with a group of m fingerprints samples of the same finger. A binary randomly generated key of "k" digits is assigned to each user as shown on figure 3.

For each fingerprint sample the FingerCode is obtained as described previously. The FingerCodes are taken as the feature vectors of p components (the number of sectors on the tessellation multiplied by the number of Gabor filters on the bank), whose values are integer numbers from 0 to 255.

The following step consists on training each SVM so that they are able to associate the feature vectors belonging to the same fingerprint of a user to its assigned binary key. Since we have only 8 samples per fingerprint, we use a subset of them for training, and the remainder for testing, as it is shown on the results section. Each SVM is trained independently of the others, since the architecture of this system consists on multiple independent subsystems with

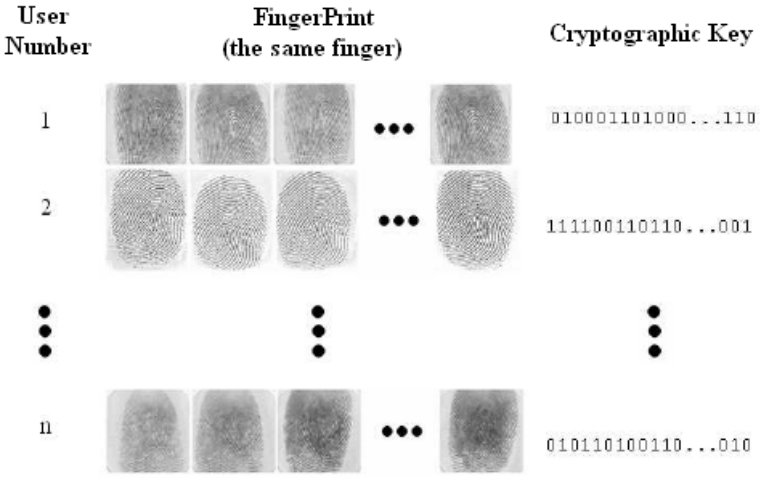


Fig. 3. m fingerprints of the same finger and cryptographic key of k digits assigned

the same characteristics, each one conformed by a SVM that associates the FingerCodes with a binary digit of the key.

It is known that the vectors belonging to the same finger are nearer among themselves than those belonging to different fingers from the same or different individuals [4]. Therefore, it is reasonable to suppose that the FingerCodes belonging to the same finger from clusters of points, in such a way that each finger would have her own region of space assigned in the space of FingerCodes. During the training step each classifier assigns a binary function that maps regions of the space of FingerCode's to the different users (each user being represented by a finger).

Possible attacks to this system they would be: 1) To make a search in the space of FingerCode's, which would be difficult because the number of possible FingerCodes is 256^p. 2) To make a search in the key, that which would be difficult because the keys are as a rule very large. 3) Through the support vectors. If the support vectors are known and the function kernel is invertible, the hyper surfaces of separation of each classifier can be known, in such a way that the response of the system can be known. This implies that the SVM models have to be encrypted too when stored on the system. 4) Since fingerprints are not confidential, someone could get a sample of it (even from a latent fingerprint), and introduce it to the system. This kind of weakness is common to most biometrics. This possibility makes this method viable only to associate public keys.

This model has a low performance, which will be show on the results section.

4.2 Model B

In this model the existence of n individuals is assumed, but the SVM are trained separately for each user, while the other $(n-1)$ work as "dummy users". For each

user a set of m fingerprints of the same finger is used, and a binary cryptographic key of k digits is assigned to this user during the training step, while the other ones are assigned a chain of k 0's. This system works by isolating the individual of interest of the rest. The SVM's would have the task of assigning a binary function to only two regions of the space, since the space would be divided exactly by the same frontier for all of them. This model performs very well reproducing the binary keys, but it is very vulnerable to attacks. However, it can be modified to for a more secure approach, as it is shown in model C.

4.3 Model C

This model solves the weakness presented by model B by randomly setting to 1 some of the bits corresponding to the other $n-1$ user or "confusion agents". When the number bits set to 1 is about half of the existent bits, this Model is equivalent in performance to model A. When only one or a small number of other bits are set to 1, this model approaches model B on performance, but without its weakness. Table 1 shows an example of assigned keys to train the array of SVMs for 6 users using model C, assigning a random key to user 4 and setting only one bit of another user in each column. This procedure has to be repeated for all the users, each user having its own array of SVM2.

Table 1. Binary code of user 4 with some of the others users bits set to 1

User	Cryptographic Code
1	000000000101000000
2	100000001000000010
3	000000000000000100
4	111100110110010001
5	001001000000000000
6	000000100010010001

The selection of the bit to change in each column for the other users is selected at random with equal probability. In this model, each set of classifiers separate the space in two regions, inside one of them is the user. The regions vary for each SVM set because of the random setting of the bit of another user in the same column. Possible attacks to this system are the same than for model A.

4.4 Security of the Models Considering the Non Confidentiality of the Biometrics

As we mentioned before, a fingerprint can be easily stolen or reproduced from a latent print, for this reason it can not be realistically considered secret or confidential. One way to override this problem is to combine the use of the fingerprint with a secret personal identification password or number. We tested a model where a random sequence was obtained using a four digits password as a seed, and this sequence was used to permute the elements on the FingerCodes.

We present this approach as model D, and we show that permuting the elements of the FingerCode does not change the performance of the key generator. This method could be used to generate private keys combining the use of a fingerprint and a password.

5 Experimental Methodology and Results

To test our architecture we selected a set of fingerprint samples from the FVC2000 and the FVC2002 databases [15]. The fingerprint selected presented good quality and where vertically aligned. The core on each fingerprint was located manually to calculate the FingerCodes. A set of eight samples per finger were selected from the FVC databases, for 30 different fingers. The architecture was tested on the four models.

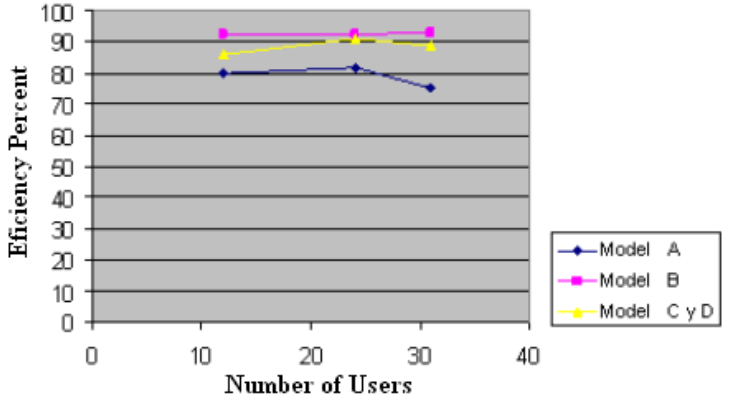


Fig. 4. Minimum value of the intervals of trust for the efficiency

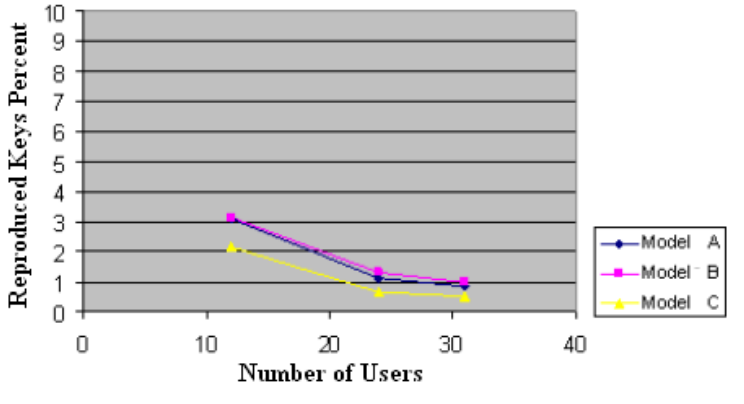


Fig. 5. Maximum value of the intervals of trust for the recovery of given valid keys by impostors

Using the classifiers like the SVM to solve the problem of the variability among FingerCodes is feasible, but is not perfect. A compromise exists between the probability of generating valid key to impostors and the probability of failing to generate the right key to a valid user. This later possibility depends greatly on the quality of the fingerprints that are used during the training and testing of the system.

We define as our measurements of efficiency the percentage of Recovery of Keys for Valid Users, and in the percentage of Recovery of Given Valid Keys by impostors or people not authorized.

Figure 4 shows the performances of the 4 models when 4 fingerprint samples per finger are used to train the SVMs and 4 for testing. An experiment was also run using 6 fingerprint samples per finger to train the SVMs and 2 for testing. While model A performance improved slightly, there were not significant changes for the other models. The results show that models C and D, which are the the most secure of the four models, are able to reproduce the correct key for approximately 90% of the presented cases.

To test the system for the generation of valid keys for not valid users, we presented the FingerCodes of 32 individuals for which the system had not been trained. The results of these experiments form models A, B and C are shown in Figure 5 with a trust interval of 95%.

The probability of generating a given valid key for impostors (a key corresponding to a given registered user) diminishes as the number of registered users increases. This probability can be reduced further by requiring a identity verification step before allowing access to the key generator.

6 Conclusions

We presented an architecture to associate binary random key to a user using FingerCodes as feature vectors and Support Vector Machines as classifiers. The architecture was tested on four models using fingerprint samples selected from the FVC2000 and the FVC2002 databases. Our models present an efficiency above 90% in generating valid key to registered users, while keeping the generation of valid keys to unregistered users low. We are currently doing research on way to improve the efficiency of the system by using other classifiers, including neural networks.

References

1. Jain A. K., Bolle R. and Pankanti S.: Biometric Personal Identification in Networked Society, Kluwer Academic Publishers, Boston/Dordrecht/London, 1999.
2. Jain A. K., Hong L. and Bolle R.: On-line Fingerprint Verification, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4) pp. 302-314, 1997.
3. Jain A. K., Ross A. and Prabhakar S., Fingerprint matching using minutiae and texture features, Proc. Int. Conference on Image Processing, vol, 3, pp.282-285, 2001.

4. Jain A. K., Prabhakar S., Hong L. and Pankanti S.: FingerCode: Filterbank-Based Fingerprint Matching, IEEE Trans. on Image Processing, Vol. 9, No.5, pp. 846-859, May 2000.
5. Boser B., Guyon I. and Vapnik V.: A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 1992.
6. Stojanov R., Gilroy Soutar G., Roberge D. and Kumar V.: Biometric encryption using image processing, SPIE, Optical Security and Counterfeit Deterrence Techniques H., 3314, pp. 178-188, 1998.
7. Cortes, C. and V. Vapnik, Support-vector network, Machine Learning, Vol. 20, pp. 273-297, 1995.
8. Furui S.: Digital Speech Processing, Synthesis, and Recognition. Mercel Dekker, inc. New York, 2001.
9. Garcia-Perera L. P., Nolazco-Flores J. A., Mex-Perera C.: Phoneme Spotting for Speech-Based Cryto-key Generation A. Sanfeliu et al. (Eds.): CIARP 2005.
10. Monrose F., Reiter M. K., Li Q. and Wetzel S.: Cryptographic Key Generation From Voice. Proceedings of the IEEE Conference on Security and Privacy, Oakland, CA. 2001.
11. Osuna E., Freund R. and Girosi F.: Support vector machines: Training and applications. Technical Report AIM-1602, MIT A.I. Lab., 1996.
12. E. Osuna, Freund R. and F. Girosi: Training Support Vector Machines: An Application to Face Recognition, in IEEE Conference on Computer Vision and Pattern Recognition, pp. 130-136, 1997.
13. Uludag U., Pankanti S., Prabhakar S. and Jain A.K.: Biometric cryptosystems: issues and challenges, Proceedings of the IEEE , Volume: 92 , Issue: 6, 2004.
14. Joachims T., SVMLight: Support Vector Machine, SVM-Light Support Vector Machine http://svmlight.joachims.org/, University of Dortmund, 1999.
15. Maltoni D., Maio D., Jain A.K., Prabhakar S., Handbook of Fingerprint Recognition, Springer, New York, 2003.
16. C. Soutar, D. Roberge, S. A. Stojanov, R. Gilroy, and B. V. K. Vijaya Kumar: Biometric Cryptosystem: Biometric encryption using image processing, in Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques II, Vol. 3314, pp. 178-188, 1998.
17. T. C. Clancy, N. Kiyavash, and D.J. Lin: Biometric Cryptosystem: Secure smartcard-based fingerprint authentication, in Proc. ACM SIGMM 2003 Multimedia, Biometrics Methods and Aplications Workshop, pp. 45-52.
18. Wan V. and Renals S.: Speaker Verification Using Sequence Discriminant Support Vector Machines IEEE Transactions on speech and audio processing, VOL. 13, NO. 2, March 2005.

Using Computational Intelligence and Parallelism to Solve an Industrial Design Problem

Fernando Asteasuain[1], Jessica A. Carballido[1], Gustavo E. Vazquez[1], and Ignacio Ponzoni[1,2]

[1] Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Av. Alem 1253, 8000, Bahía Blanca, Argentina
[2] Planta Piloto de Ingeniería Química - CONICET
Complejo CRIBABB, Camino La Carrindanga km.7 CC 717, Bahía Blanca, Argentina
{fa, jac, gev, ip}@cs.uns.edu.ar

Abstract. In this work we present a critical analysis of three novel parallel-distributed implementations of a multi-objective genetic algorithm (pdGAs) for instrumentation design applications. The pdGAs aim at establishing a sensible configuration of sensors for the initialization of instrumentation design studies of industrial processes. They were built on the basis of an evolutionary island model, the master-worker paradigm, and different migration and parameter control policies. The performance of the resulting implementations was assessed by testing algorithmic behavior on an industrial example that corresponds to an ammonia synthesis plant. The three pdGAs' results were highly satisfactory in terms of speed-up, efficiency and instrumentation quality, thus revealing to constitute competitive tools with strong potential for their use in the industrial area. As well, from an overall point of view, the pdGA version with adaptive parameter control represents the best implementation's alternative.

Keywords: Genetic Algorithms, Distributed Computing, Instrumentation Design.

1 Automatic Initialization for Instrumentation Design Studies

The sensor network design (SND) aims at determining the most convenient amount, distribution and kind of measuring devices to be installed in an industrial plant so that effective, efficient and complete monitoring is achieved without unnecessary expenditure. Industrial plants consist of a series of items of equipment, such as reactors, interconnected by means of streams. Each process unit can be modeled by a non-linear system of algebraic equations that represents its functioning under steady state operating conditions. A widespread approach to carry out the SND is based on a judicious analysis of the state variables (e.g., the reactor's temperature) that appear in the complete mathematical model derived from plant topology. The central idea is to reduce the number of measurements by estimating the value of some variables from model equations [1].

An initial set of measurements should be provided to begin this procedure. Thus, process variables can be classified into *measured variables*, i.e., those whose value will

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 188 – 197, 2006.

always be available from the information provided by the measuring devices, and *unmeasured variables*, which need to be estimated, otherwise remaining unknown. Once the initialization has been defined, an **observability analysis (OA)** is performed to determine the subset of unmeasured variables that could be calculated from model equations. In other words, the unmeasured variables are classified into *observable* and *unobservable* (or *indeterminable*) *variables* [1]. Then, the degree of process knowledge that can be achieved is inversely proportional to the number of unobservable variables. The variables that are associated with crucial information about the industrial process are called critical. An ideal sensor configuration should have no critical *indeterminable* variables. Therefore, whenever the classification reveals that crucial process variables will remain unknown, the engineer has to modify the proposed configuration by adding a few sensors that allow either the measurement or the estimation of the variables of interest. The OA thus becomes an iterative procedure.

Ponzoni *et al* [2] concluded that the structural approach constitutes the best alternative to carry out the observability analysis. Nevertheless, some structural methods are based on combinatorial searches, which demand long execution times. Then, as each OA sweep is costly, any improvement in the initialization procedure resulting in a reduction of the number of OA iterations would be extremely advantageous.

The search for a satisfactory initialization constitutes a combinatorial optimization problem, which implies finding a trade-off solution that simultaneously fulfils contradicting goals. As regards quality, the initialization algorithm should tend to produce configurations that provide maximum sensor reliability and degree of observability, as well as minimum installation and operating costs. In addition, just like any feasible initialization technique, the procedure should be as fast as possible in terms of execution times. This combinatorial optimization problem can be solved by means of a multi-objective genetic algorithm (GA), which constitutes an excellent choice that enables an adequate representation and treatment of all the objectives, also yielding useful solutions relatively quickly [3]. In this sense, a new OA procedure was proposed in Carballido *et al.* [4], where the initial sensor configuration is obtained through a GA. The GA suggests several good potential alternatives, and then, the expert user selects one of them to start up the OA.

Main objective. Since this new approach achieved good results for small and medium size chemical processes reported in [4], we decided to test the method with more complex industrial plants. In this sense, it is important to notice that in this problem the search space grows exponentially with the number of unmeasured variables, that quantity increasing together with the complexity of the mathematical model of the process under study. Then, the search space for the GA in real situations becomes really huge. Moreover, the execution time required to calculate the fitness function grows in a polynomial manner as the number of variables increases [4]. For all these reasons, arose the idea of distributing the GA's computational effort between various workstations thereafter evaluating the impact in the treatment of complex industrial problems.

In this sense, in recent works it has been demonstrated that well designed parallel distributed (pdGAs) manage to perform well in much lower execution times than their sequential counterparts [5-8]. In particular, the *multiple-deme parallel model*, more commonly known as the island model, has achieved excellent results in a wide spectrum of applications [5, 8, 9]. Besides the simplicity of the approach, which exploits in a

straightforward way the underlying parallelism of the GAs, most authors agree in the fact that there exist various key issues to establish in order to ensure that a pdGA based in this model will succeed. Between those aspects they emphasize: type of the islands - homogeneous or heterogeneous -, the way parameters are implemented into each island – fixed or adaptive – and the migration schemata – synchronous or asynchronous - [8].

Considering these topics, and after performing an exhaustive revision on today's main trends [6, 8-12], we decided to implement three parallel-distributed variants of the GA proposed in Carballido *et al.* [4]. By means of an experimental analysis carried out on complex and huge study cases, we aim at determining which the best choice is, also comparing them with their sequential counterpart.

This article is organized as follows: section 2 presents a brief description of the sequential GA proposed in Carballido *et al.* [4]; section 3 describes the design of the parallel-distributed GA (pdGA) and the main distinctive features of the three variants; section 4 depicts the experimental work and results; the final section discusses the conclusions.

2 The Sequential Genetic Algorithm for Sensor Initialization

In this section we present the most relevant features of the sequential GA on which the parallel versions presented in this article are based. This GA is founded on the implementation introduced in Carballido *et al.*'s work [4], where more details can be found.

Input information. The main input of the GA for OA initialization is the occurrence matrix O built from the steady state mathematical model of the plant under study. The rows and columns of this binary matrix correspond to model equations and variables, respectively. A nonzero in row i column j means that in equation i appears variable j. The GA also needs information about the cost and reliability associated to the instrument that would be required in order to "measure each process variable" (e.g., the cost associated with the instrument needed to measure the reactor's temperature). The cost of measuring a variable is calculated as the price of the instrument plus its installation cost. The reliability is associated with that of the instrument that measures the variable. These values are stored in $\boldsymbol{N}$-dimensional vectors, where $\boldsymbol{N}$ is the number of variables.

Representation, operators and selection. Binary strings are used to represent the individuals, each string of length $\boldsymbol{N}$ standing for a feasible sensor configuration. A nonzero value in one of the bits means that the variable at that position should be measured. Other data are saved along with the genotype, but for the sake of simplicity the symbol $\boldsymbol{i}$ is used to represent either the whole individual or its genotype, depending on the context. Since the representation is the canonical one, the mutation and crossover operators need not be redefined. The bit-flip method is implemented for the mutation, and the one-point approach is used for the crossover. The selection method is the binary tournament.

Dynamic mutation probability. Even though the method selected to perform the mutation is the classic one, the probability used to apply the operator does not stay fixed during the evolution process. Mutation probability is first initialized with the value $1/\boldsymbol{N}$, and as the population evolves this value is decreased. The schema adopted to manage this probability is based on several studies which confirm that parameter control

succeeds at improving the GA's performance [12, 13]. The idea behind this statement is that the mutation probability should decrease as evolution takes place, thus exploiting the good individuals that where already found.

Non feasible individuals. An individual is considered non feasible if it contains a non-zero in a position that represents a variable that cannot be physically measured, such as an enthalpy. The non feasible individuals are treated differently at each stage of the GA. At first, the initial population is randomly generated imposing the restriction that the positions that correspond to non measurable variables are always initialized with a zero. Also, as new gene information is only introduced through the mutation operator, those positions are considered non-mutable. It is important to observe that the non-mutable variables can not be erased from the chromosome since they have to be present when the observability term of the fitness function is calculated.

2.1 The Fitness Function

The algorithm aims at finding the individual ***i*** that simultaneously exhibits the best performance with respect to cost, reliability and observability. Therefore, there is a trade-off that involves the following three objective functions: minimize cost, and maximize both reliability and observability.

Cost. The total cost of an individual is the sum of the values of all the elements in *cv* - a cost vector with length N which holds the price of measuring each variable - that correspond to nonzero entries in *i*.

$$\boldsymbol{C}\,(\boldsymbol{i}) = \sum_{j=1}^{N} (\boldsymbol{cv}[j] * \boldsymbol{i}[j]) \ . \tag{1}$$

Reliability Following the same line of reasoning, the reliability is calculated as:

$$\boldsymbol{R}\,(\boldsymbol{i}) = \sum_{k=1}^{N} (\boldsymbol{rv}[k] * \boldsymbol{i}[k]) \ . \tag{2}$$

In Equation 2, ***rv*** is a reliability vector that contains information about the accuracy of the instrument to be installed in order to measure each variable.

Observability. In contrast with the other two objective functions, this one cannot be calculated in a straightforward manner. Its estimation is based on the mathematical operation called Forward Triangularization (FT). Details of the algorithm that implements the FT can be found in [2]. It returns estimates on the number of unmeasured variables that can be directly calculated by solving individual equations from the system of equations, given the measurements defined in *i*. In short, the value returned by this function is:

$$\boldsymbol{Ob}\,(\boldsymbol{i}) = \mathrm{FT}\,(\boldsymbol{i}) \ . \tag{3}$$

The FT algorithm is in fact the core of the rigorous algorithm for OA, which also includes modules with more detailed analysis tools that refine the FT results at the expense of much higher computing times. In view of its short run times, the FT constitutes a convenient procedure for initialization purposes.

The Aggregating Approach. The construction of the fitness function requires a normalization criterion to reconcile the $C(i)$, $R(i)$ and $Ob(i)$ values, judiciously combining them so that none is undervalued. The standard procedure consists in normalizing each objective in the [0, 1] range. Thus,

$$F(i) = NR(i) + NOb(i) + 1 - NC(i) . \quad (4)$$

where $NC(i)$, $NR(i)$ and $NOb(i)$ are the three normalized objectives. As it can be noticed, $F(i)$ ranges between 0 and 3, 3 being the number of objectives.

At this point it is important to notice that, whereas nowadays the trend is to use Pareto dominancy techniques to tackle multi-objective problems, it is well known that the aggregating approach outperforms those techniques in combinatorial optimization [3] in which this problem application is categorized.

3 The Parallel-Distributed Genetic Algorithm

Parallel models succeed at increasing efficiency and providing a means of making a better use of computational resources. Moreover, they are highly advantageous when applied to genetic algorithms, since GAs are inherently parallel [6-8, 10]. In this sense, parallel genetic algorithms (pGAs) exhibit very good numerical performance due to the use of a well-distributed population [7, 14]. Furthermore, Van Veldehuizen *et al.* [11] established the advantages of employing pGAs for the treatment of multi-objective problems in which our problem classifies.

In particular, in the *island mode*l, the population is distributed in several groups called "islands", where each subpopulation evolves separately in parallel. In this scheme the subpopulation on each island normally consists of many individuals, where occasionally some of them migrate to another island. For this reason, the island model deals with several parameters associated to the migration, such as the number of migrants per migration cycle, the migration cycle frequency, the choice and replacement criterion for migrants and the topology that connects the existing subpopulations [5, 9, 11]. Other relevant features are the degree of heterogeneity between the control parameters (crossover and mutation probability) in each island, and whether these parameters are fixed or adapted during the evolution of the subpopulations [9, 13].

3.1 Our Proposal

From a literature review [6, 8-13], we decided to implement models with *heterogeneus control parameters* as they preserve population diversity better than the homogeneus schemata [9,13]. Concerning the migration policy, we explored two variants: *synchronic and asyncronic migration with fixed evolution parameters*, and for the *synchronic case* we also implemented a version with adaptive parameters. In short, we implemented the following three parallel alternatives of the Carballido *et al.*'s GA [4]:

s-pdGA: heterogeneous islands with fixed parameters and synchronic migration.
a-pdGA: heterogeneous islands with fixed parameters and asynchronic migration.
ad-pdGA: heterogeneous islands with adaptive parameters and synchronic migration.

The use of adaptive parameters together with asynchronic migration was discarded because previous studies discouraged this alternative [9]. The distributed-processing strategy follows the master-worker paradigm. The parallel-distributed implementation consists in decomposing the population into subsets, each of them handled by a different worker who executes the sequential GA within his subpopulation. A migration policy determines how to exchange individuals between islands, thus making it possible to share genetic material. The role of the master is to control and synchronize the computations carried out by the workers. The communication was implemented on a local area network (LAN) using the Parallel Virtual Machine library [15]. PVM was adopted in view of its reliability and its capacity to handle heterogeneous resources as if they were a single parallel machine.

Software Architecture. All the versions share the same general architecture that follows the traditional master-worker scheme. The master creates as many workers as required by the user and coordinates their activities, giving them information on the problem's parameters. Each worker receives the parameters, generates the population at random, and executes the GA. After each cycle, all the workers send the master their best individuals. This information is used by the master to stop the evolution according to the stopping criterion that will be explained later.

Migration. During an evolutionary cycle some individuals are exchanged between islands. There is a *migration interval* because migrations are carried out periodically every g generations, where g is a fixed value. The migration may be synchronic (**s-pdGA**) or asynchronic (**a-pdGA**), depending on whether the workers have to wait for their colleagues in order to coordinate the migrants' exchange or not. The interconnection structure that models the migration flow between workers follows a ring topology, which proved to be efficacious in other applications [5, 9, 16]. In brief, each worker migrate his best individual to the next worker in the ring. Then, each worker replaces his worst individual by the new immigrant.

Role of the Master. The Master first creates the slaves and distributes the parameters such as crossover and mutation parameters. After that, he cycles until either a maximum number of iterations is reached or a "solution" is found. The master considers that a "solution" was found according to the following criterion: in every cycle he receives the best individual from each worker, picks out the best, and compares it against a *potential optimal solution*. If the error is lower than a preset bound he terminates and kills the workers. It is important in this regard to notice that the *potential optimal solution* is obtained either from the literature or from previous experimental studies. This termination policy is of fundamental importance because, if the GA is stopped when solutions of similar quality have been achieved, then a fair comparison between the sequential and parallel GA versions can be assured ([5], [6], [17]).

Role of the Worker. Each worker receives the values for the parameters and creates his own population at random. Then, he goes into a cycle that closely resembles a GA's cycle, the main difference arising at the end of each generation where migration takes place and the best individual is sent to the master. For efficiency reasons this cycle is implemented as infinite. A worker will stop only on the master's command, this order being issued once the termination condition has been fulfilled.

Adaptive Parameters. The implementation of the **ad-pdGA** includes the notion of adaptive parameters in accord with the Tongchim and Chongstitvatana's work [9], which was successfully tested on different applications. This version was generated in order to introduce self-adjustment capabilities during the search. On every island the population is split, and each half is associated to a different *set of parameters* (PS) that contains the probabilities of crossover and mutation. A fitness value is associated with each PS, whose value is calculated as the average fitness of the corresponding subpopulation.

Migration of Adaptive Parameters. For a given island in the **ad-pdGA**, both PSs are initialized at random at the beginning of the procedure. The range for the values: [0.0, 0.2] for the mutation and [0.6, 0.8] for the crossover, proved to be adequate for initialization purposes. At the moment of migration, each worker receives the immigrant together with his neighbor's best PS and its associated fitness. If the fitness of the incoming PS is better than that of the best local PS, both PSs are crossed over and mutated in order to yield two new PSs that will immediately become the parameters for the two subpopulations of that island. This migration is synchronic. The representation and operators for the PSs are based on the real numbers' representation explained in Michalewicz [18].

4 Performance Analysis for an Industrial Case

Industrial Test Example. The algorithmic performance was assessed by carrying out the instrumentation analysis of an industrial plant whose main features are described in Bike [19]. The plant produces 1500 ton/day of anhydrous liquid ammonia at 240 K and 450kPa with a minimum purity of 99.5%. The product is obtained by means of the Haber-Bosch process, which consists in a medium-pressure synthesis in a catalytic reactor followed by an absorption procedure that removes the ammonia with water. The rigorous mathematical model of this plant, which is required to build the occurrence matrix **O**, was generated using the ModGen software [20]. The resulting system contains 557 non-linear algebraic equations and 546 process variables.

Experimental Environment. The computational experiments were performed on Pentium IV PCs of 2.4 GHz and 512 MB RAM, connected through a Fast Ethernet 100Mb/s network. The pdGAs were executed on 1, 2, 4, and 8 processors. The population size for the sequential version was set at 128 individuals, while the subpopulation sizes for the parallel runs were divided proportional to the number of islands (processors). The total number of runs executed for the different models amounted to 100, while the migration interval was set at five generations, this empirical value being adopted after carrying out several preliminary tests.

Heterogeneous islands were employed in the asynchronic (**a-pdGA**) and synchronic (**s-pdGA**) versions. For each version, exploitation prevailed in half of the islands, with high crossover probabilities (75%) and less genetic diversity, tending to reduce the mutation probability (10%). For the other half, exploration was favored by lowering the crossover probability (65%), simultaneously increasing the mutation probability (15%). In all cases, the probability of mutation diminished dynamically with the number of generations, as explained in section 2. The pdGAs stop the evolution either when an individual whose fitness value is at least 2.62 is found, or after 300 generations have elapsed. This termination threshold was selected from previous experimental studies [4].

Analysis of the Results. Table 1 shows the average speed-up and efficiency values obtained with the three pdGA versions. Whereas in the cases of 2 and 4 workers the adaptive version outperformed the other two; the synchronic implementation achieved better results in the 8-workers scheme. The scalability of the parallel system was always satisfactory. The high speed-up results we obtained are not really surprising since, in the literature, the capacity of parallel distributed AGs to achieve good levels of performance is well known [7, 14]. At this point it is important to clarify that, even tough we are aware that to improve the quantification of the system's scalability it would have been interesting to experiment with 16 and 32 workstations, for infrastructure limitations we were forced to use a maximum of 8 processors.

Table 1. pdGA Speed-up and Efficiency

Number of Workers	PopSize	Synchronic		Asynchronic (a-pdGA)		Adaptive (ad-pdGA)	
		Speed-Up	Efficiency	Speed-Up	Efficiency	Speed-Up	Efficiency
2	64	1.87	93%	1.84	92%	1.93	96%
4	32	3.90	97%	3.17	79%	3.93	98%
8	16	7.79	97%	7.26	90%	7.0	87%

Regarding the quality of the solutions (fitness values) attained by the sequential and parallel versions, table 2 reflects the proportion of runs in which the fitness of the best individual was above the threshold shown in column 1. Due to the heterogeneity of the island model, tests were performed on two versions of the sequential GA, respectively containing the exploration and the exploitation parameters, thus yielding to fairer comparisons. It is clear that the pdGAs succeeded in finding high-quality solutions more regularly than their sequential counterparts. In turn, from a comparison of the parallel versions' performance it can be concluded that it improves as the number of islands augments. This behavior was expectable since as the number of islands increases, thus yielding to lower sizes for the populations, so do the selective pressure [9]. In this way the best individual on each island is replicated more times as a consequence of migration. Then, the convergence of the pdGA is accelerated towards solutions with higher fitness values.

Table 2. Solutions' Quality for the Sequential and Parallel Versions

Thresh-olds	Sequential		2 Workers			4 Workers			8 Workers		
	Explor.	Explot.	s-pdGA	a-pdGA	ad-pdGA	s-pdGA	a-pdGA	ad-pdGA	s-pdGA	a-pdGA	ad-pdGA
>2,62	4%	0%	5%	4%	14%	24%	48%	26%	71%	88%	54%
>2,61	8%	0%	11%	16%	34%	36%	70%	56%	85%	98%	78%
>2,60	12%	12%	30%	38%	50%	63%	84%	74%	93%	100%	86%
>2,59	24%	28%	55%	70%	64%	80%	94%	88%	96%	100%	92%
>2,58	42%	54%	73%	80%	80%	89%	98%	98%	99%	100%	98%

The values attained for the different thresholds were included in order to make qualitative comparisons among the parallel versions (see figure 1). It can be observed that the **a-pdGA** achieves the most satisfactory performance in most of the cases, followed by the **ad-pdGA** version. All in all, in our opinion the adaptive version exhibited the best overall performance since it surpasses the asynchronic version in terms of computing times without significant detriment in the solution's quality.

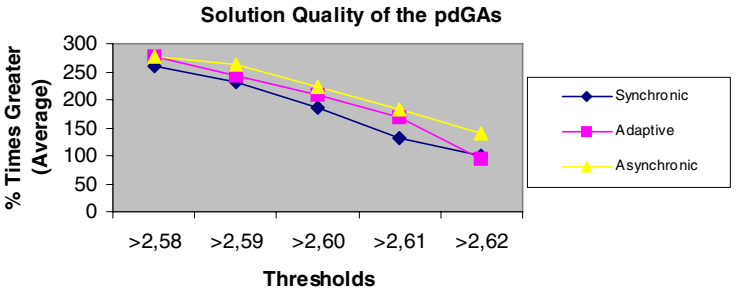


Fig. 1. Quality comparison between the pdGAs versions

Finally, it is important to note that the sensor configurations yielded by the pdGAs were always satisfactory from the engineering standpoint. In other words, all the resulting classifications met practical industrial requirements since all the critical variables were tagged as observable. These results encourage the use of the pdGA on real industrial applications.

5 Conclusions and Future Work

In this work we present a comparative study of three parallel implementations of a multi-objective genetic algorithm (pdGA) developed to obtain the best sensor initialization for industrial plant's instrumentation design. The pdGA's versions use an evolutionary island model with master-worker architecture, and the implementation was carried out with the PVM message-passage library. The algorithmic functioning was tested by means of the selection of a suitable configuration of sensors on a complex ammonia synthesis plant. All the pdGA's results were excellent in comparison with their sequential counterparts, not only as regards the quality of the resulting instrumentation but also in terms of speed-up and efficiency. Besides, by contrasting the behavior of the parallel versions implemented in this work, it can be concluded that the use of adaptive parameters is advantageous since it leads to improvements in the global algorithmic performance.

Although these results are encouraging, there are aspects still to be considered so as to complete the analysis. The main one concerns with testing the relationship between the different migration intervals' length and the number of migrants per migration cycle, in order to optimize these parameters. In this respect, we are now working on the implementation of an adaptive global migration scheme that will augment the concurrence level of the pdGA. Finally, we are assessing the pdGA's performance on other real-life industrial problems with various degrees of complexity so as to corroborate, in a wider spectrum, the results we have already obtained.

Acknowledgment

Authors would like to acknowledge for their economic support to ANPCyT: Grants Nº11-12778 (BID 1728/OC-AR) and Cod. 917.

References

1. Romagnoli, J.A., Sánchez, M.C.: Data Processing and Reconciliation for Chemical Process Operations. Academic Press (1999)
2. Ponzoni, I., Sánchez, M.C., Brignole, N.B.: A New Structural Algorithm for Observability Classification. Ind. Eng. Chem. Res., 38, 8 (1999) 3027-3035
3. Coello Coello, C.A.: A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems. 1, 3 (1999) 269–308
4. Carballido, J.A., Ponzoni, I., Brignole, N.B.: A Novel Application of Evolutionary Computing in Process Systems Engineering. In: Raidl, Günther R.; Gottlieb, Jens (Eds.) Evolutionary Computation in Combinatorial Optimization. Lecture Notes in Computer Science, Vol. 3448. Springer-Verlag Berlin Heidelberg New York (2005) 12-22
5. Sena, G.A., Megherbi, D., Isern, G.: Implementation of a Parallel Genetic Algorithm on a Cluster of Workstations. FGCS 17 (2001) 477-488
6. Alba, E., Tomassini, M.: Parallelism and Evolutionary Algorithms. IEEE Trans. Evolutionary Comput. 6 (2002) 443-462
7. Cantú-Paz, E., Goldberg, D.E.: Predicting speedups of idealized bounding cases of parallel genetic algorithms. 7 Int. Conf. on Genetic Algorithms, Morgan Kaufmann, (1997) 338-345
8. Cantú-Paz, E., Goldberg, D.E.: Efficient Parallel genetic algorithms: theory and practice. Comput. Methods Appl. Mech. Engrg. 186 (2000) 221-238
9. Tongchim, S., Chongstitvatana, P.: Parallel Genetic Algorithm with Parameter Adaptation. Information Processing Letters. 82 (2002) 47-54
10. Alba, E., Troya, J.M.: Analysing synchronous and asynchronous parallel distributed genetic algorithms. FGCS 17 (2001) 451-465
11. Van Veldhuizen, D.A., Zydallis, J.B., Lamont, G.B.: Considerations in engineering parallel multiobjective evolutionary algorithms. IEEE Trans. Evolutionary Comput. 7 (2003) 144-173
12. Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary Comp.: Comments on the History and Current State. IEEE Trans. Evolutionary Comput. 1, 1, (1997) 3-17
13. Eiben, Á.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Trans. Evolutionary Comput. 3, (1999) 124-141
14. Alba, E.: Parallel evolutionary algorithms can achieve super-linear performance. Information Processing Letters. 82, (2002) 7-13
15. Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Mancheck, R., Sunderam, V.: PVM: Parallel Virtual Machine. A users guide and tutorial for network parallel computing. MIT Press (1994).
16. Pereira, C., Lapa, C.: Parallel island GA applied to a nuclear power plant auxiliary feedwater system surveillance tests policy optimization. Annals of Nuclear Energy. (2003) 1665–1675.
17. Safe, M., Carballido, J.A., Ponzoni, I., Brignole, N.B.: On Stopping Criteria for Genetic Algorithms. In: Bazzan, A., Labidi, S. (Eds.): Advances in Artificial Intelligence - SBIA 2004. Lecture Notes in Artificial Intelligence, Vol. 3171. Springer-Verlag, Berlin Heidelberg New York (2004) 405–413
18. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd Ed. Springer-Verlag, Berlin Heidelberg New York (1996)
19. Bike, S.: Design of an Ammonia Synthesis Plant, CACHE Case Study. Department of Chemical Engineering, Carnegie Mellon University, (1985)
20. Vazquez, G.E., Ponzoni, I., Sánchez, M.C., Brignole, N.B.: ModGen: A Model Generator for Instrumentation Analysis. Advances in Engineering Software. 32 (2001) 37-48

Two-Phase GA-Based Model to Learn Generalized Hyper-heuristics for the 2D-Cutting Stock Problem

Hugo Terashima-Marín[1], Cláudia J. Farías-Zárate[1], Peter Ross[2], and Manuel Valenzuela-Rendón[1]

[1] Tecnológico de Monterrey, Center for Intelligent Systems
Monterrey, Nuevo León 64849 Mexico
{terashima, a00789069, valenzuela}@itesm.mx
[2] Napier University, School of Computing
Edinburgh EH10 5DT UK
P.Ross@napier.ac.uk

Abstract. The idea behind hyper-heuristics is to discover some combination of straightforward heuristics to solve a wide range of problems. To be worthwhile, such combination should outperform the single heuristics. This paper presents a GA-based method that produces general hyper-heuristics that solve two-dimensional cutting stock problems. The GA uses a variable-length representation, which evolves combinations of condition-action rules producing hyper-heuristics after going through a learning process which includes training and testing phases. Such hyper-heuristics, when tested with a large set of benchmark problems, produce outstanding results (optimal and near-optimal) for most of the cases. The testebed is composed of problems used in other similar studies in the literature. Some additional instances of the testbed were randomly generated.

1 Introduction

Cutting stock is a problem widely studied because it has many practical applications. The problem belongs to the class of most difficult problems known as NP-hard [6]. Given a set of pieces, the problem is to generate cutting patterns from sheets of stock material, or objects, that optimize the number of objects used. In this particular investigation problems involve only 2D-rectangular pieces. Since many precise requirements and constraints vary from industry to industry, many different approaches and techniques have been proposed for solving the problem [11]. The aim of this paper is to explore a novel alternative on the usage of evolutionary approaches to generate hyper-heuristics when solving 2D-rectangular cutting stock problems. A hyper-heuristic is used to define a high-level heuristic that controls low-level heuristics [3]. The hyper-heuristic should decide when and where to apply each single low-level heuristic, depending on the given problem state. In recent work which is based on the research by Ross *et al.* [17] on unidimensional binpacking, evolutionary approaches have been used to generate hyper-heuristics for the 2D-Regular Cutting Stock Problems. Terashima

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 198–207, 2006.

et al. [18] use the XCS Classifier System to generate the hyper-heuristics. In other related work [19], authors use a Genetic Algorithm with integer and fixed-length representation to produce hyper-heuristics. Both previous approaches assemble a combination of single heuristics (selection and placement), and deliver very competitive results for a set of different problem instances, beating in fact, results produced by the single heuristics. The investigation in this article presents a method to generate general hyper-heuristics intended to solve a wide variety of instances of 2D-regular cutting stock problems. The procedure learns a hyper-heuristic by going through a training phase using instances with a variety of features. The generated hyper-heuristic is tested later with a collection of unseen examples providing excellent results. The general method is based on a variable-length Genetic Algorithm, where the chromosome is conformed of a series of blocks, representing condition-action rules.

The paper is organized as follows. Section 2 describes the cutting stock problem. Section 3 presents the solution method proposed. Section 4 discusses the experiments, and their results. Finally, in section 5 we include our conclusions.

2 The Cutting Stock Problem

The Cutting Stock Problem (CuSP) is among the earliest problems in the literature of operational research. In 1939, Kantorovich studied obvious applications in all the industries whose products were in a flat sheet form; this research was published in 1960 [13]. Since then, there have been many investigations on the problem: an abstract description of the different solution methods which have been given to the problem [9]; the applications and solutions to the CuSP problem [5]; and the solution methods of the problem [4]. Due to the diversity of problems and applications, Dyckhoff [5] has proposed a very complete and systematic categorization of cutting and packing problems. His survey integrates a general system of 96 problems for the Cutting Stock with four main features and their subtypes as follows: Dimensionality: One (1), Two (2), Three (3) or n; Assignation form (All the larger objects and a selection of small figures (B), or A selection of large objects and all the small figures (V)); Assortment of large objects (One object (O), Identical shapes (I), or Different Shapes (D)); and Assortment of small figures (Few figures of different shapes (F), Many figures of different shapes (M), Many figures of few of different and incongruent shapes (R), or Congruent shapes (C)). Then our work will be limited to a 2VIC-Cutting Stock Problem.

3 Solution Approach

This section describes the set of heuristics used in this investigation and explains in detail the model proposed.

The Set of Heuristics Used

In a one dimensional packing problem, the related heuristics refer to the way the pieces are selected and the bins in which they will be packed. For a two

dimensional problem such as the 2VIC-CuSP, additional difficulty is introduced by defining the exact location of the pieces inside the object. In this investigation two kinds of heuristics were considered: for selecting the figures and objects, and for placing the pieces into the objects. Some of the heuristics were taken from the literature, others were adapted, and some other variations developed by us. We chose the most representative heuristics in its type, considering their individual performance presented in related studies and also in an initial experimentation on a collection of benchmark problems. The selection heuristics used in this research are: Next Fit (NF), First Fit (FF), Best Fit (BF), Worst Fit (WF), Almost Worst Fit (AWF), First Fit Decreasing (FFD), Next Fit Decreasing (NFD), and Djang and Fitch (DJD). These heuristics are described in detail by Ross *et al.* [17] and Hopper *et al.* [11]. The placement heuristics belong to the class of bottom-left heuristics, that is, they keep the bottom-left stability in the layout. They are based on a sliding technique. The placement heuristics we used are: Bottom-Left (BL) [12] and Improved-Bottom Left (BLLT) [14]. From these, two new heuristics were generated to consider rotation in the piece to place. These heuristics are called BLR and BLLTR. There are 40 combinations of selection and placement heuristics and they are shown in Table 1.

Table 1. Representation of actions

Actn	Selection	Placement	Actn	Selection	Placement
1	First Fit (FF)	BL	21	Next Fit Decreasing (NFD)	BL
2		BLR - BL Rotate	22		BLR - BL Rotate
3		BLLT - Improved BL	23		BLLT - Improved BL
4		BLLTR - Improved BLR	24		BLLTR - Improved BLR
5	First Fit Decreasing (FFD)	BL	25	Best Fit (BF)	BL
6		BLR - BL Rotate	26		BLR - BL Roate
7		BLLT - Improved BL	27		BLLT - Improved BL
8		BLLTR - Improved BLR	28		BLLR - Improved BLR
9	First Fit Increasing (FFI)	BL	29	Best Fit Decreasing (BFD)	BL
10		BLR - BL Rotate	30		BLR - BL Roate
11		BLLT - Improved BL	31		BLLT - Improved BL
12		BLLTR - Improved BLR	32		BLLTR - Improved BLR
13	Filler+FFD	BL	33	Worst Fit (WF)	BL
14		BLR - BL Rotate	34		BLR - BL Rotate
15		BLLT - Improved BL	35		BLLT - Improved BL
16		BLLTR - Improved BLR	36		BLLTR - Improved BLR
17	Next Fit (NF)	BL	37	Djang and Finch (DJD)	BL
18		BLR - BL Rotate	38		BLR - BL Rotate
19		BLLT - Improved BL	39		BLLT - Improved BL
20		BLLTR - Improved BLR	40		BLLTR - Improved BLR

Combining Heuristics with the Proposed GA

Hyper-heuristic is motivated by the objective to provide a more general procedure for optimization [3]. Meta-heuristics methods usually solve problems by operating directly on the problem. Hyper-heuristics deal with the process to choose the right heuristic for solving the problem at hand. The idea is to discover a combination of simple heuristics that can perform well on a whole range of problems and in such a way that a heuristic's strengths make up for the drawbacks of another. In this research we use a GA [10, 7] with variable length chromosomes, a resemblance of what is called a *messy*-GA [8]. The solution model proposed carries features from work by Ross *et al.* [16], for uni-dimensional bin-packing problems. In our research, a GA with variable-length individuals is proposed to find a combination of single heuristics (selection and placement) to solve efficiently a wide variety of instances of 2D-Regular cutting stock problems. The

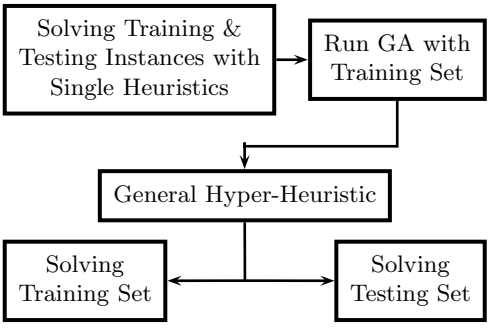


Fig. 1. Solution Model

basic idea is that, given a problem state P, this is associated with the closest point in the chromosome which carries the selection and placement to be applied. The chromosome therefore represents a complete recipe for solving a problem, using a simple algorithm: until the problem is solved, (a) determine the current problem state P, (b) find the nearest point to it, (c) apply the heuristic attached to the point, and (d) update the state. The GA is looking for the chromosome (representing a hyper-heuristic) which contains the rules that apply best to any intermediate state in the solving process of a given instance. The instances are divided into two groups: the training and the testing set. The general procedure consists in solving first all instances in both sets with the single heuristics (each is a combination of a selection and a placement heuristic). This is carried out to keep the best solution that is later used also by the GA we propose. The next step is to let the GA work on the training set until termination criterion is met and a general hyper-heuristic is produced. All instances in both the testing and training sets are then solved with this general hyper-heuristic. The complete process is presented in Figure 1.

Representation. Each chromosome is composed of a series of *blocks* where each block j includes six numbers. The first five represent an instance of the problem state. The initial four numbers indicate the percentage of pieces that remain to be packed according to the following categories (Ao: object area, Ap: item area): h_j huge items ($Ao/2 < Ap$), l_j large items ($Ao/3 < Ap \leq Ao/2$), m_j medium items ($Ao/4 < Ap \leq Ao/3$), and s_j small items ($Ap \leq Ao/4$). The fifth number, r_j, represents the percentage of the total items that remain to be packed. The sixth number indicates the combination of heuristics (selection and placement), associated with this instance. For a given problem state, the inital five numbers would lie in a range between 0 and 1, so that the actual problem state is a point inside the unit five-dimensional space. Nevertheless, we allow the points defined in each block to lie outside the unit cube, so we redefined the range to be from -3 to 3. The idea was to allow points to lie outside the $[0, 1]^5$ hypercube. If you don't, then consider some problem state represented by a point close to a corner of the cube. In that case, its nearest point is likely to be somewhere further inside the cube, and the region of space

closest to that point is therefore likely to be large. If points near the edge of the hypercube need special handling in some way that most other points do not, then maybe they can be handled by some heuristic attached to a point outside the cube. At each step, the algorithm applies that heuristic that is associated to the block that is closest to actual problem state. We measure the distance d between the problem state P' and the instance inside each block j with the formula $d = \sqrt{(h_j - h')^2 + (l_j - l')^2 + (m_j - m')^2 + (s_j - s')^2 + (r_j - r')^2}$ where each term indicates the square of the difference for each category previously defined. The action was selected from all possible combinations of selection and placement heuristics shown in Table 1.

The Fitness Function. The quality of solution produced by either a single heuristic or a hyper-heuristic for a given instance, is based on the percentage of usage for each object u given by $P_u = \frac{\sum_{j=1}^{n} Ap_j}{Ao}$, where Ap_j represents the area of item j; Ao the object area; and n the number of items inside the object. Then, the fitness is given by

$$FF = \frac{\sum_{u=1}^{No} P_u^2}{No} \tag{1}$$

where No is the total number of objects used, and P_u is the percentage of utilization for each object u. Now, how is the fitness of a chromosome measured during the GA cycle? To do this, first, it is necessary to compute the fitness produced by each individual combination of selection and placement heuristics, for each instance. The best heuristic combination and its result, for each specified instance i are stored (let us call it BSH_i). These results are prepared in advance of running the GA. The steps of the GA (steady-state type) cycle are: (1) Generate initial population; (2) Assign 5 problems (random) to each chromosome and get its fitness; (3) Apply selection (tournament), crossover and mutation operators to produce two children; (4) Assign 5 problems (random) to each new child and get its fitness; (5) Replace the two worst individulas with the new offspring; (6) Assign a new problem to every individual in the new population and recompute fitness; and, Repeat from step 3 until a termination criterion is reached. To compute the fitness for each chromosome (at steps 2 and 4 of the above cylce), the distance between the solution obtained by that individual with respect to the best result given by the single heuristic (BSH_i) is measured. The fitness is a weighted average and it is given by: $FF(HH) = \frac{\sum_{k=1}^{5} P_k \cdot (FF_k - BSH_k)}{\sum_{k=1}^{5} P_k}$ where P_k is the number of times the k-th assigned instance has been tackled so far; BSH_k is the best fitness obtained for the k-th assigned instance by a single heuristic; and FF_k is the fitness obtained by the hyper-heuristic for the k-th assigned instance (using equation 1).

After each generation l, a new problem is assigned to each individual m in the population and its fitness is recomputed by a weighted average as follows: $FF_m^l = \frac{FF_m^{l-1} \cdot mp_m + FF(m)}{mp+1}$ where FF_m^{l-1} is the fitness for individual m in the previous generation; mp_m is the number of problems individual m has seen so

Table 2. Experiment Type I: Hyper-heuristic (HH-I) produced by the GA With group A as training set

Huge	Large	Medium	Small	Remain	Actn	Huge	Large	Medium	Small	Remain	Actn
0.37	0.08	0.87	−1.26	0.07	37	1.08	−1.44	−0.49	−1.54	−2.83	25
0.12	0.19	0.87	0.07	1.55	37	1.08	−1.44	−0.49	−1.26	1.37	25
0.12	0.19	0.18	1.11	0.81	13	1.28	−0.55	−0.13	0.07	−0.51	12
0.12	−0.57	0.18	1.11	0.59	15						
−0.39	−0.24	0.87	−1.26	1.37	25						
0.71	−0.54	−0.56	0.17	−0.96	34						

far; $FF(m)$ is the fitness obtained by individual m for the new problem and computed with the fitness function given by equation 1.

4 Experiments and Results

This section presents the experiments carried out during the investigation and the results obtained. Problems from several sources have been used in this research. Part of the benchmark set was taken from the literature (the OR-Library [1], set by Martello and Vigo [15], set by Berkey and Wang [2], set by Terashima *et al.* [18]), and the rest is composed of randomly generated instances for which an optimal solution is known. The collection includes 540 different instances. They were divided into two groups (A and B) of 270 instances each (chosen at random). Aiming at testing the model effectiveness, three kinds of experiments were carried out. We used a population size of 50 during 400 generations, crossover probability of 1.0, and mutation probability of 0.05. Overall, each instance was solved, in average, 89 times in the training phase.

Experiment Type I
In this experiment, the instance set A is used as the training set. After this phase is complete, a general hyper-heuristic is produced by the GA, which is shown in Table 2. We call it HH-I. It includes 9 rules indicating the different problem states and the associated action or heuristic to be applied. In the resulting hyper-heuristic, we can observe that despite the same single heuristic appears more than once in the action part, the conditions for applying it are quite different.

Next, we solved independently instances in set A and set B using HH-I. Table 3 (top part) summarizes results for set A. Results are compared against those generated by the best single heuristics. Heuristics are grouped in subsets of four where the selection heuristic is common among them. Figures in cells indicate the percentage of problems solved with a particular number of extra objects (left column) when compared with the results provided by the best heuristic. For example, on the column labeled 'FFD', heuristics 5 to 8 solved 67.96% of the instances in set A with the same number of objects as the best heuristic for each given instance. It is interesting to see, if we focus in the results obtained by HH-I, that 80.37% of problems were solved with no extra objects (percentage higher than any other set of heuristics). In 1.48% of problems, HH-I obtained solutions with one object less than the best heuristic. It is important to em-

Table 3. HH-I: Number of extra objects for problems in set A and set B

Extra Objects for problems in set A											
		FF	FFD	FFI	Filler	NF	NFD	BF	BFD	WF	DJD
Obj.	HH-I	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36	37-40
-1	**1.48**										
0	**80.37**	30.3	67.96	16.85	73.7	30.55	66.29	30.74	69.25	30.37	**74.81**
1	18.14	31.4	29.44	21.85	23.51	29.81	30.92	30.55	27.77	31.85	22.59
2		22.2	2.40	7.96	2.59	22.22	2.59	22.96	2.77	21.85	2.40
3		9.07	0.18	9.07	0.18	10.37	0.18	8.70	0.18	9.25	0.18
4		5.55		8.14		5.55		5.37		5.37	
5		1.11		9.81		1.29		1.48		1.11	
> 5		0.18		26.29		0.18		0.85		0.18	

Extra Objects for problems in set B											
		FF	FFD	FFI	Filler	NF	NFD	BF	BFD	WF	DJD
Obj.	HH-I	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36	37-40
-1	**4.81**										
0	**76.92**	33	66.75	20.46	71.01	33.98	67.5	33.61	69.53	33.24	**71.66**
1	18.26	30.5	29.94	23.79	24.62	29.25	27.77	29.62	26.11	30.37	20.92
2		15.4	2.4	7.96	3.92	14.16	3.79	15.09	3.42	15.46	1.66
3		9.62	0.83	7.68	0.37	10.74	0.37	10.55	0.37	9.25	0
4		8.7	0.92	7.03	0.55	8.51	0.55	7.96	0.55	9.07	0.18
5		1.85	0.92	7.96		2.59		2.59		1.03	
> 5		0.74	1.20	25.09		0.74		0.55		0.55	

phasize that the best single heuristic is not always the same for all instances, but there is a tendency to achieve reasonable performance from those heuristics known to be good, in general, for problems of this kind. 'DJD' is in this case, and it comes second to HH-I. However, other sets of heuristics show questionable performance since some of them need more than five extra objects to solve a high percentage of instances. For set B, whose instances were not seen before by HH-I, results are shown in Table 3 (bottom part). Again, the hyper-heuristic shows better performance than the best single heuristics with 76.92% problems with 0 extra objects and 4.81% of problems with one object less. The other groups of heuristics behave very much the same as their performance shown with set A.

Experiment Type II

In order to confirm the performance and roboustness of the model, other set of experiments were designed. Basically, the procedure is the same as Experiment Type I, but in this case, set B is used as the training set instead. The hyper-heuristic created is shown in Table 4. We label it HH-II. HH-II comprises nineteen blocks. Another interesting observation is that the single heuristics composing the HH-II are different from those composing HH-I. Heuristic 13 (Filler+FFD+BL), however, is common in both, and appears in four different rules in the hyper-heuristic. Heuristic 28 (BF+BLLTR) appears also four times, and Heuristic 7 (FFD+BLLT) appears three times with rather different conditions.

Table 5 (top part) shows results for HH-II for instances in set B. Percentage on solved problems within zero and one less extra bin is 83.69% (77.40%+6.29%). Only in 15.81% of problems, the hyper-heuristic uses one extra bin than the best single heuristic. Overall, results are better than the subsets of heuristics. Table 5 (bottom part) presents results for instances in set A when using hyper-heuristic HH-II. These instances had not seen before in the training phase. Results confirm the acceptable performance of the hyper-heuristic.

Table 4. Experiment Type II: Hyper-heuristic (HH-II) produced by the GA

Huge	Large	Medium	Small	Remain	Actn	Huge	Large	Medium	Small	Remain	Actn
-0.95	−1.09	0.26	1.67	−0.02	8	0.63	−1.54	0.55	0.14	−0.82	7
0.34	0.14	0.1	1.92	−0.65	28	0.86	0.27	0.45	0.86	0.71	39
-0.28	−0.59	0.88	−0.38	1.32	13	0.16	−0.14	−0.6	−2.14	−0.34	28
−1.07	−0.41	0.45	−0.38	−0.72	15	0.63	−1.54	0.55	0.15	−0.14	28
−1.07	−0.41	−0.82	−0.35	−0.34	14	0.63	−1.54	0.55	0.14	−0.82	7
0.24	−0.38	0.88	−0.38	1.32	13	0.24	−0.38	−1.9	1.7	−0.25	7
−0.28	−0.59	−1.6	−0.86	−0.5	2	−1.02	−1.15	0.4	−0.88	0.98	21
−0.86	0.27	0.26	−0.38	1.32	13	−0.04	1.71	1.35	−0.19	−0.76	13
−0.99	−1.61	−1.61	1.92	−0.65	28	0.14	−0.79	−0.16	0.86	0.71	39
0.26	−0.5	−0.77	1.21	−0.28	3						

Table 5. HH-II: Number of extra objects for problems in set B and set A

Extra Objects for problems in set B											
		FF	FFD	FFI	Filler	NF	NFD	BF	BFD	WF	DJD
Obj.	HH-II	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36	37-40
-1	**6.29**										
0	**77.40**	31.2	69.62	17.59	72.7	32.22	69.25	31.85	71.29	31.48	**74.81**
1	15.81	30.1	28.14	22.59	25	28.88	28.14	29.25	26.48	30	23.14
2		16.8	2.03	8.33	2.03	15.55	2.40	16.48	2.03	16.85	1.85
3		10.1	0	8.51	0	11.11	0	10.92	0	9.62	0
4		9.07	0.18	7.77	0.18	8.88	0.18	8.33	0.18	9.44	0.18
5		1.85		8.88		2.59		2.59		1.03	
> 5		0.74		26.29		0.74		0.55		0.55	

Extra Objects for problems in set A											
		FF	FFD	FFI	Filler	NF	NFD	BF	BFD	WF	DJD
Obj.	HH-II	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36	37-40
-1	**2.85**										
0	**77.77**	30	67.77	16.85	73.51	30.37	66.11	30.37	69.07	30	**74.62**
1	18.51	32.2	29.25	21.85	23.3	30.18	30.74	31.11	27.59	32.59	22.40
2		21.8	2.59	7.96	2.77	22.03	2.77	22.96	2.96	21.48	2.59
3		9.7	0.18	9.07	0.18	10.37	0.18	8.51	0.18	9.25	0.18
4		5.5		8.14		5.55		5.37		5.37	
5		1.11		9.81		1.29		1.48		1.11	
> 5		0.18		26.29		0.18		0.18		0.18	

Table 6. HH-I and HH-II: Number of extra objects for problems in set C

			Heuristics									
			FF	FFD	FFI	Filler	NF	NFD	BF	BFD	WF	DJD
Obj.	HH-I	HH-II	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36	37-40
-1	**2.59**	**2.96**										
0	**79.6**	**79.6**	68.1	75.37	46.66	74.44	67.40	74.07	66.48	73.88	67.59	77.59
1	17.7	17.4	31.4	24.25	40.92	25.19	32.22	25.55	33.51	25.74	32.03	22.03
2			0.37	0.37	11.66	0.37	0.37	0.37		0.37	0.37	0.37
3					0.74							

Experiment Type III

We generated additional 270 instances (group C) with the intention to compare performance of both hyper-heuristics generated in the two previous experiments. Table 6 presents the results. In addition, every single instance was solved using each of the combinations of selection and placement heuristics, keeping the best combination for each instance. It can be observed that behavior in both hyper-heuristic is very similar. Both solve around 3% of problems with one object less than the best heuristic, around 80% of problems with zero extra objects, and approximately 18% of problems with one extra object.

Looking at the results, it is clear in all cases, that the method to form hyper-heuristics, and the hyper-heuristics themselves are efficient, at least with respect to the number of objects used for each instance. However, it is important to

Table 7. Summary of Results: Hyper-heuristics, Best Single Heuristics and Testing Sets

Problems	Results								
	HH-I			HH-II			BSH		
Extra Objects	-1	0	1	-1	0	1	0	1	> 1
Set A	1.48	80.37	18.14	2.85	77.77	18.51	74.81	22.4	2.58
Set B	4.81	76.92	18.26	6.29	77.4	15.81	71.66	20.92	1.84
Set C	2.59	79.6	17.7	2.96	79.6	17.4	77.59	22.03	0.37

get a better feeling of the real advantages or the proposed approach, and the practical implications of using it. For example, regarding the computational cost for delivering solutions by our approach, it is slightly higher than the time used by the simple heuristics which run in just few seconds. Table 7 summarizes results on the performance of hyper-heuristics HH-I and HH-II with respect to the different testing sets we used (A, B, and C). Around 82% of problems in all sets are solved with zero or one less extra objects, whereas the best single heuristics solve only around 75% of problems with zero extra objects.

5 Conclusions and Future Work

This document has described experimental results in a two-phase GA model which evolves combinations of condition-action rules for solving 2D-Regular cutting stock problems. These combinations are called hyper-heuristics. Overall, the scheme identifies efficiently general hyper-heuristics after going through a learning procedure with training and testing phases. When applied to unseen examples, those hyper-heuristics solve most of the problems very efficiently, in fact, much better than the best single heuristic for each instance. Ideas for future work involve extending the proposed strategy to solve problems including other kinds of pieces such as polygonal, irregular, and multidimensional.

Acknowledgments

This research was supported in part by ITESM under the Research Chair CAT-010 and the CONACYT Project under grant 41515.

References

1. J. E. Beasley. Beasley operations research library. *Collection of problems for 2D packing and cutting*, 2003.
2. J. O. Berkey and P. Y. Wang. Two-dimensional finite bin packing algorithms. *Journal of Operational Research Society*, 38:423–429, 1987.
3. E. Burke, E. Hart, G. Kendall, J. Newall, P. Ross, and S. Schulenburg. Hyper-heuristics: An emerging direction in modern research technolology. In *Handbook of Metaheuristics*, pages 457–474. Kluwer Academic Publishers, 2003.
4. C. H. Cheng, B. R. Fiering, and T. C. Chang. The cutting stock problem. a survey. *International Journal of Production Economics*, 36:291–305, 1994.

5. H. Dyckhoff. A topology of cuting and packing problems. *European Journal of Operation Research*, 44:145–159, 1990.
6. M. Garey and D. Johnson. *Computers and Intractability.* W.H. Freeman and Company, New York, 1979.
7. D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Adison Wesley, 1989.
8. D. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis and first results. *Complex Systems*, pages pp. 93–130, 1989.
9. B. L. Golden. Approaches to the cutting stock problem. *AIIE Transactions*, 8:256–274, 1976.
10. J. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, 1975.
11. E. Hopper and B. C. Turton. An empirical study of meta-heuristics applied to 2D rectangular bin packing. *Studia Informatica Universalis*, pages 77–106, 2001.
12. S. Jakobs. On genetic algorithms for the packing of polygons. *European Journal of Operations Research*, 88:165–181, 1996.
13. L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6:366–422, 1960.
14. D. Liu and H. Teng. An improved bl-algorithm for genetic algorithm of the orthogonal packing of rectangle. *European Journal of Operations Research*, 112:413–419, 1999.
15. S. Martello and D. Vigo. Exact solution of the two-dimensional finite bin packing problem. *Dipartimento di Elettronica, Informatica e Sistematica*, 1998.
16. P. Ross, J. M. Blázquez, S. Schulenburg, and E. Hart. Learning a procedure that can solve hard bin-packing problems: A new ga-based approach to hyper-heuristics. *Proceedings of GECCO 2003*, pages 1295–1306, 2003.
17. P. Ross, S. Schulenburg, J. M. Blázquez, and E. Hart. Hyper-heuristics: learning to combine simple heuristics in bin-packing problems. *Proceedings of GECCO 2002*, pages 942–948, 2002.
18. H. Terashima-Marín, E. J. Flores-Álvarez, and P. Ross. Hyper-heuristics and classifier systems for solving 2D-regular cutting stock problems. *Proceedings of the Genetic and Evolutionary Computation Conference 2005*, pages 637–643, 2005.
19. H. Terashima-Marín, A. Morán-Saavedra, and P. Ross. Forming hyper-heuristics with GAs when solving 2D-regular cutting stock problems. *Proceedings of the Congress on Evolutionary Computation*, pages 1104–1110, 2005.

Mirrored Traveling Tournament Problem: An Evolutionary Approach

Fabrício Lacerda Biajoli and Luiz Antonio Nogueira Lorena

Instituto Nacional de Pesquisas Espaciais - INPE
Laboratório Associado de Computação e Matemática Aplicada - LAC
Av. dos Astronautas, 1.758 - São José dos Campos-SP, Brazil

Abstract. The *Mirrored Traveling Tournament Problem* (mTTP) is an optimization problem that represents certain types of sports timetabling, where the objective is to minimize the total distance traveled by the teams. This work proposes the use of hybrid heuristic to solve the mTTP, using an evolutionary algorithm in association with the metaheuristic Simulated Annealing. It suggests the use of Genetic Algorithm with a compact genetic codification in conjunction with an algorithm to expand the code. The validation of the results will be done in benchmark problems available in literature and real benchmark problems, e.g. Brazilian Soccer Championship.

1 Introduction

Scheduling problems in sports has become an important class of optimization problems in recent years. The professional sport leagues represent a major economic activity around the world. For several sports, e.g. soccer, basketball, football, baseball, hockey, etc, where the teams plays a double round-robin tournament among themselves, where the games are played in different places during some time period, the automating of that schedulings are necessary and very important. Other facts fortify the application of optimization techniques. Teams and leagues do not want to waste their investments in players and structure in consequence of poor scheduling of games; sports leagues represent significant sources of revenue for radio and television world networks; the scheduling interfere directly in the performance of the teams; etc. On the other hand, sport leagues generate extremely challenging optimization problems, that attract attention of the Operational Research communities.

The scheduling problems in sports are known in the literature as Traveling Tournament Problem and it was proposed by Easton et al. [7]. The TTP abstract the salient features of Major League Baseball (MLB) in the United States and was established to stimulate research in sport scheduling. Since the challenge instances were proposed the TTP has raised significant interest. Several works in different contexts (see e.g. [2],[4],[9],[12],[13],[14],[18]) tackled the problem of tournament scheduling in different leagues and sports, which contains many interesting discussion on sport scheduling. Basically, the schedule of MLB is a conflict between minimizing travel distances and feasibility constraints on the

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 208–217, 2006.

home/away patterns. A TTP solution is a double round-robin which satisfies sophisticated feasibility constraints (e.g. no more than three away games in a road trip) and minimizes the total travel distances of the teams.

Problems of that nature contain in general many conflicting restrictions to be satisfied and different objectives to accomplish, like minimize the total road trips of the teams during the tournament, one just game per day and per team, accomplishment of certain games in stadiums and in pre-established dates, number of consecutive games played in the team's city and out, etc. To generate good schedulings, satisfying all constraints, is a very hard task. The difficulty of solution of that problem is attributed to the great number of possibilities to be analyzed, e.g., for a competition with 20 teams there are $2,9062 \text{ x } 10^{130}$ possible combinations ([3]).

This work proposes the application of evolutionary techniques and local search for solving the mirrored version of the TTP, known as Mirrored Traveling Tournament Problem - mTTP ([13]). This paper is organized in five sections, being this the first. In the next section, the traveling tournament problem and your mirrored version are described. In section three, the methodology is detailed, with neighborhoods and the algorithm implemented. Section four presents the computational results and section five describe some conclusions of this paper.

2 Problem Description

The *Traveling Tournament Problem* was first proposed by Easton et al. in [7]. A scheduling to a *double round-robin* (DRR) tournament, played by n teams, where n is a even number, consists in a schedule where each team plays with each other twice, one game in its home and other in your opponent's home. A game between teams T_i and T_j is represented by unordered pair (i, j). That schedule needs $2(n-1)$ rounds to represents all games of the tournament. The input data consists of the number of teams (n), a symmetric matrix D, n x n, where D_{ij} represents the distance between the home cities of the teams T_i and T_j.

The cost of a team is the total distance traveled starting from its home city and return there after the tournament ending. The cost of the solution is the sum of the cost of every team.

The objective is to find a schedule with minimum cost, satisfying the following constraints:

- No more than three consecutive home or away games for any team;
- A game of T_i at T_j's home cannot be followed by the game of T_j at T_i's home;

The *Mirrored Traveling Tournament Problem* (mTTP) proposed by Ribeiro and Urrutia in [13] is a generalization of TTP that represents the common structure in Latin-America tournaments (e.g. Brazilian Soccer Championship). The main difference is the concept of *mirrored double round-robin* (MDRR). A MDRR is a tournament where each team plays every other once in the $n-1$ rounds, followed by the same games with reversed venues in the last $n-1$ rounds.

The objective is the same of TTP, find a schedule with minimum cost satisfying the same constraints plus an additional constraint: the games played in round R are the same played in round $R + (n - 1)$ for $R = 1, 2, ..., n - 1$, with reversed venues.

3 Methodology

The methodology used to solve the problem is based on the use of Genetic Algorithms in association with the metaheuristic Simulated Annealing. The idea is to use the Genetic Algorithm as construction phase, generating new solutions starting from the individuals' crossing and the Simulated Annealing to improve the local search in those new solutions.

The representation of a schedule is a table indicating the opponents of the teams, where each line corresponds to a team and each column corresponds to a round. The opponent's representation is given by the pair (i, j), where i represents the team T_i and j represents the round r_j (e.g., the opponent of the team T_1 in round r_2 is given by $(1, 2)$). If (i, j) is positive, the game takes place at T_i's home, otherwise at T_i's opponent home.

In this work only the $n - 1$ first rounds (*first half*) are represented, because the $n - 1$ last rounds (*second half*) are the mirror with reversed venues and all alteration in the first half affects the second half (see Figure 1).

	First half					Second half				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

Fig. 1. Representation of a Schedule

The rest of that section describes the neighborhood and the algorithms implemented in this work.

3.1 The Neighborhood

Three different moves have been defined to compose distinct kinds of neighborhood, named *Home-away swap*, *Team swap* and *Games swap*, from a schedule S. The neighborhood of a schedule S is the set of the schedules (feasibles and infeasibles) which can be obtained by applying one of three types of moves.

Home-away swap. This move swaps the home/away roles of a game involving the teams T_i and T_j. The application of the move *Home-away swap* in a solution S obtain a solution S', with a single game swapped, by reversing the game's place. In other words, if team T_i plays at home with T_j (T_j plays away) in S, then T_j plays at home and T_i plays away in S'.

The figure 2 shows a schedule before and after the application of *Home-away swap* move. In this example, the move swap the home/away roles of the game involving the teams T_1 and T_4.

	First half					Second half				
T_j/r_k	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

	First half					Second half				
T_j/r_k	1	2	3	4	5	6	7	8	9	10
1	-6	-4	2	-5	-3	6	4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	1	6	2	-5	3	-1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

Fig. 2. Schedule before (left) and after (right) the application of *Home-away swap*

Team swap. This move swaps the schedule of two teams, T_i and T_j. Only the games where T_i and T_j play against each other are not swapped.

	First half					Second half				
T_j/r_k	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

	First half					Second half				
T_j/r_k	1	2	3	4	5	6	7	8	9	10
1	-6	-4	2	-5	-3	6	4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	1	6	2	-5	3	-1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

Fig. 3. Schedule before (left) and after (right) the application of *Team swap*

The figure 3 shows the application of *Team swap* move in teams T_3 and T_5.

Games swap. This move consists in selecting an arbitrary game and enforcing it to be played in a round, followed by the necessary modifications to avoid teams playing more than one game in the same round. In consequence, more games would be swapped to maintain the feasible of the schedule. The modifications that have to be applied to the current schedule give rise to an ejection chain move. Ejection chains are based on the notion of generating compound sequences of moves by linked steps in which changes in selected elements cause other elements to be ejected from their current state, position, or value assignment ([13]).

	First half					Second half				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

	First half					Second half				
T_i/r_k	1	2	3	4	5	6	7	8	9	10
1	**-5**	**-3**	2	**4**	**-6**	**5**	**3**	-2	**-4**	**6**
2	**-6**	**4**	-1	**-5**	**3**	**6**	**-4**	1	**5**	**-3**
3	-4	**1**	-5	**6**	**-2**	4	**-1**	5	**-6**	**2**
4	3	**-2**	6	**-1**	**5**	-3	**2**	-6	**1**	**-5**
5	**1**	-6	3	**2**	**-4**	**-1**	6	-3	**-2**	**4**
6	**2**	5	-4	-3	**1**	**-2**	-5	4	3	**-1**

Fig. 4. Schedule before (left) and after (right) the application of *Games swap*

The figure 4 shows the schedule produced by the application of *Game swap* move. Note that several games are swapped, when the team T_3 was enforcing to play with team T_1 in round r_2.

3.2 The Algorithm

In this work was implemented a Genetic Algorithm ([10]) that uses the Simulated Annealing ([11]) metaheuristic to address new individuals to a local optimum. The application of local search in the individuals can be related with the combination of learning and evolution (Baldwin effect, [17]). In general, the learning is a search for the near viable solution and the modifications will be incorporate for the individual. The use of the SA metaheuristic leaves the stage of local search more aggressive, resulting in individuals more and more adapted inside of the population.

A compact representation of the chromosomes (individuals) was proposed for the application of the GA. The chromosomes are submitted to an algorithm of code expansion, which decodes them in scales of games.

In this representation, each gene of the chromosomes is associated to a team. The figure 5 presents an example of the compact representation used. Once the

3	2	5	1	6	4

Fig. 5. Example of a chromosome for a tournament with 6 teams

chromosome is defined, then it is submitted to an algorithm of code expansion, the well known Polygon Method ([5]), in association with heuristic to definition of the home/away roles. All solutions generated by this phase are feasible.

Consider a vector V of size n (n even) where each position i is associated to a team. The execution of the polygon method starts with the definition of the base team. This team is positioned in the first position of the vector V. The other teams are positioned in the rest of the positions, $i = 2, .., n$. In each round $r_i = 1, ..., n-1$ the base team plays with the team of the position 2. The teams of positions $i = 3, ..., (n/2) + 1$ plays with the teams of positions $n - i + 3$ (n is the number of teams). Defined all games of a round r_i, the teams of position $i = 3, ..., n$ are moved to the position $i - 1$ and the team of position 2 are moved

Round 1	**3**	2	5	1	6	4
Round 2	**3**	5	1	6	4	2
Round 3	**3**	1	6	4	2	5
Round 4	**3**	6	4	2	5	1
Round 5	**3**	4	2	5	1	6

Fig. 6. Polygon Method

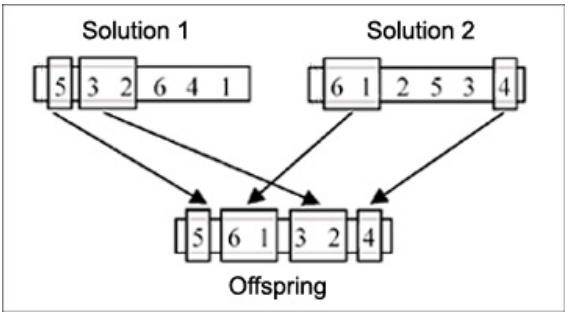


Fig. 7. BOX Crossover

to position n. The base team are not moved. The figure 6 presents an example of execution.

The initial population is randomly defined and each individual is submitted to a *Randomized Non-Ascendent* method (RNA), with the *home-away* swap move only .

A reproduction mechanism, based on evolutionary processes, is applied on the population to explore the search space and to find better solutions for the problem. The operator of crossover implemented was based on the Block Order Crossover (BOX, [15]), illustrated in the Figure 7 . The "parents" are combined, through the random copy of both individuals blocks, what results in a new offspring, containing the two parents' genetic information.

In each generation a constant number of individuals are selected. Two individuals are randomly selected in the population for each crossover, which produces only one new individual. This offspring can, eventually, suffer mutation. The mutation used was the *Games Swap* move (see fig. 4).

Immediately after the execution of the genetic operators, the algorithm of code expansion is executed for generation the schedule.

The figure 8 presents the pseudo code of the algorithm implemented.

The metaheuristic Simulated Annealing, with *Home-away swap* and *Team swap* moves, are applied in each offspring, to address it to a local optimum.

The idea of use the methods above mentioned in association is related to the fact that evolutionary algorithms find some difficulties to treat optimization

```
procedure GA
1  Initialize Population P;
2  while (g < nGenerations) do
3     while (i < nIndividuals) do
4        parents ← Select two individuals ∈ P;
5        offspring ← Crossover(parents);
6        if (MutationAcceptionCriterion) then
7           Mutation(offspring);
8        end-if
9        offspring' ← LocalSearch(offspring);
10       Evaluate(offspring');
11       Add offspring' in P;
12       i ← i + 1;
13    end-while
14    P ← Select(P);
15    g ← g + 1;
16 end-while
end-GA
```

Fig. 8. Pseudo Code of the implemented GA

problems with constraints. On the other hand, isolated application of optimization techniques based in constraints and local search can find more difficulties in virtue of low quality local optimum.

The application of evolutionary algorithms together with local search is justified for two main aspects:

- The evolutionary process will act just in the compact codification, always feasible. Therefore not reducing your effectiveness;
- The existence of a candidates population for the solution in evolution, together with the application of a local search procedure, increases the chances of obtaining local optimum of good quality.

4 Experiments and Computational Results

The algorithm was coded in C++ and was run on *Pentium IV* 3.0 *GHz clock with* 512 *Mbytes of RAM memory.*

The benchmark's instances, described in [7] and adapted to the mirrored form in [13] was used to validate the results. A real-life instance (br2003.24), where 24 teams playing in the main division of the 2003 edition of Brazilian Soccer Championship was also tested. These test problems are available to download in http://mat.gsia.cmu.edu/TOURN/. The parameters of the methods were empirically chosen, after several simulations. The population starts with 100 individuals and 50 offsprings are generated in each one of the 10 generations. The mutation probability and crossover probability are 30% and 100% respectively.

Table 1 shows the results for the considered instances. For each instance is reported the best known solution ([13]), the obtained by this approach, the

Table 1. Computational results

Instances	Best known	Obtained	*gap*(%)	Time(*secs.*)
circ4	20	20	0%	2
circ6	72	72	0%	4
circ8	140	142	1,41%	48
circ10	240	282	14,89%	365
circ12	456	458	0,44%	51
circ14	714	714	0%	26
circ16	980	1014	3,35%	264
circ18	1306	1370	4,67%	604
circ20	1882	1890	0,42%	28
nl4	8276	8276	0%	2
nl6	26588	26588	0%	3
nl8	41928	43112	2,75%	55
nl10	58190	66264	12,18%	130
nl12	120655	120981	0,27%	317
nl14	208086	208086	0%	140
nl16	279618	290188	3,64%	142
br2003.24	503158	511256	1,58%	938

relative gap in percent between the best and obtained solutions, and the total computation times in seconds.

The results presented demonstrate that the proposed methodology can be competitive, because using only three movements was possible to obtain values near of the best results known in the literature, getting to reduce to zero the gap in six of the seventeen instances and being near of reducing to zero in nine others. On the other hand, the results of circ10 and nl10 wasn't good, demonstrating the requirement of improving the algorithm, mainly the neighborhoods. This fact is easily explained. The schedule generated by this implementation is fixed in a pattern, due the polygon method characteristics. The ejection chains moves are very important, because they are able to find solutions that are not reachable through other neighborhoods. Admissible moves in other neighborhoods may appear after an ejection chain move is performed, in situations where none existed before. If they are not used, algorithms based on local improvement strategies may easily stop at local optimum of low quality.

The result of the real-life instance was very interesting. First, because it is the larger instance in the literature, with 24 teams. Second, because was observed a reduction of 51.2% in the total distance traveled, where in the official schedule the teams traveled 1.048.134 km and in the schedule found by the work they traveled 511.256 km only.

5 Conclusions

In this work was investigate the Mirrored Traveling Tournament Problem, first published in [13], with a implementation of a Genetic Algorithm in association with the metaheuristic Simulated Annealing.

A compact representation for the chromosomes (population individuals) was proposed. These chromosomes are submitted to a code expansion algorithm, used to decodes them in scales of games. Initially, the home-away roles are aleatory defined. After this phase a *Randomized Non-Ascendent* method (RNA) is applied in the schedule to address it to a local optimum. This method showed effectiveness, because all solutions found are feasible, increasing the chance of obtained local optimum with good quality.

Three different neighborhood structures for local search was investigated: two simple neighborhood (*Home-away swap* and *Team swap*) and a more complicated based on ejection chain, whose the importance was described in section three. The results show the need of explores other neighborhoods to escape from the pattern imposed to the solutions generated by the Polygon Method.

The approach become very promising, when the reported results are analyzed. Seventeen benchmark instances was tested and in six the algorithm got to reduce to zero the gap and almost reduced to zero in nine others. One real-life instance, the 2003 edition of Brazilian Soccer Championship, was also tested, with reduction of 51.2% of the total distance traveled by the official schedule.

Finally, this work explores the mirrored instances of TTP, because its represents common structure in Latin-America tournaments (e.g. Brazilian Soccer Championship). There are a variety of open issues that need to be addressed, e.g., to consider a large neighborhood to obtain high-quality solutions. In the same way the championships has many other restrictions that should be analyzed: treatment of the classic games, allocation of stadiums, do not consider metric distance, but the airfares, and many others real constraints.

References

1. Anagnostopoulos, A., Michel, L., Van Hentenryck, P., Vergados, Y.: A Simulated Annealing Approach to the Traveling Tournament Problem. Proceedings of Cpaior'03 (2003)
2. Biajoli, F. L., Chaves, A. A., Mine, O. M., Souza, M. J. F., Pontes, R. C., Lucena, A., Cabral, L. F.: Scheduling the Brazilian Soccer Championship: A Simulated Annealing Approach. Fifth International Conference on the Practice and Theory of Automated Timetabling, Patat2004, Pittsburgh, USA, (2004) 433-437
3. Concílio, R., Zuben, F. J.: Uma Abordagem Evolutiva para Geração Automática de Turnos Completos em Torneios. Revista Controle e Automação. Vol. 13, n.2, (2002) 105-122
4. Costa, D.: An Evolutionary Tabu Search Algorithm and the NHL Scheduling Problem. Infor. Vol. 33, n. 3, (1995) 161-178
5. Dinitz, J., Lamken, E., Wallis, W. D.: Scheduling a Tournament. In C.J. Colbourn and J. Dinitz, editors, Handbook of Combinatorial Designs, CRC Press, (1995) 578-584
6. Dowsland, K. A.: Simulated Annealing. C.R. Reeves, Editor, Modern Heuristic Techniques for Combinatorial Problems, Advanced Topics in Computer Science Series, Chapter 2. Blackwell Scientific Publications, London, (1993) 20-69
7. Easton , K., Nemhauser, G., Trick, M.: The Traveling Tournament Problem Description and Benchmarks. Seventh International Conference on the Principles and Practice of Constraint Programming, CP'99, (2001) 580-589

8. Goldberg, D. E.: Genetic Algorithms in Search. Optimization and Machine Learning. Addison-Wesley: Berkeley, 223 pag., (1989)
9. Hamiez, J. P., Hao, J. K.: Solving The Sports League Scheduling Problem with Tabu Search. Lecture Notes In Artificial Intelligence 2148, Springer (2001), 24-36
10. Holland, J. H.: Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press, 211 pag., (1975)
11. Kirkpatrick, S., Gellat, D. C., Vecchi, M. P.: Optimization by Simulated Annealing. Science, Vol. 220, (1983) 671-680
12. Nemhauser, G., Trick, M.: Scheduling a Major College Basketball Conference. Operations Research, Vol. 46(1), (1998) 1-8
13. Ribeiro, C.C., Urrutia S.: Heuristics for the Mirrored Traveling Tournament Problem. Fifth International Conference on the Practice and Theory of Automated Timetabling, Patat2004, Pittsburgh, USA, (2004) 323-342
14. Schönberger, J., Mattfeld, D. C., Kopfer, H.:Memetic Algorithm Timetabling for Non-Commercial Sport Leagues. European Journal of Operational Research, Vol. 153(1), (1989) 102-116
15. Syswerda, G.: Uniform Crossover in Genetic Algorithms. International Conference on Genetic Algorithms (ICGA), Vol. 3, Virginia, USA,(1989) 2-9
16. Trick, M. A.: A Schedule-Then-Break Approach to Sports Timetabling. Lecture Notes In Computer Science, Vol. 2079, (2000) 242-253
17. Whitley, D., Gordon, V. S., Mathiask K.: Lamarckian Evolution, the Baldwin Effect and Function Optimization. In Davidor, Y., Schwefel, H.-P. and Männer, R. (eds.) Proceedings of the Third Conference on Parallel Problem Solving from Nature, BerlSpringer, October, (1994) 6-15
18. Zhang, H.: Generating College Conference Basketball Schedules by a Sat Solver. Proceedings Of The Fifth International Symposium on the Theory and Applications of Satisfiability Testing, Cincinnati, (2002) 281-291

Pattern Sequencing Problems by Clustering Search

Alexandre C.M. Oliveira[1] and Luiz A.N. Lorena[2]

[1] Universidade Federal do Maranhão - UFMA, Departamento de Informática, São Luís MA, Brazil
acmo@deinf.ufma.br

[2] Instituto Nacional de Pesquisas Espaciais - INPE, Laboratório Associado de Computação e Matemática Aplicada, São José dos Campos SP, Brazil
lorena@lac.inpe.br

Abstract. Modern search methods for optimization consider hybrid search metaheuristics those employing general optimizers working together with a problem-specific local search procedure. The hybridism comes from the balancing of global and local search procedures. A challenge in such algorithms is to discover efficient strategies to cover all the search space, applying local search only in actually promising search areas. This paper proposes the Clustering Search (*CS): a generic way of combining search metaheuristics with clustering to detect promising search areas before applying local search procedures. The clustering process aims to gather similar *information* about the problem at hand into groups, maintaining a representative solution associated to this information. Two applications to combinatorial optimization are examined, showing the flexibility and competitiveness of the method.

Keywords: Hybrid search metaheuristic, pattern sequencing problem, Clustering search.

1 Introduction

Local search methods have been combined with search metaheuristics in different ways to solve particular problems more efficiently. Hill-climbing procedures are largely employed in the so called memetic algorithms (MA) as a Lamarckian learning process [1]. For example, a simple crossover can work as a local search around the parents, hill-climbing by repeatedly generating some number of offspring and replacing the worst parent [2].

The main challenge in such hybrid methods is to define efficient strategies to cover all search space, applying local search only in actually promising areas. Elitism plays an important role towards achieving this goal, once the best solutions represent promising neighborhood. However, such well-evaluated solutions can be concentrated in few areas and thus the exploitation moves are not rationally applied.

An approach attempting to find out relevant areas for continuous optimization is a parallel hill-climber, called Universal Evolutionary Global Optimizer

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 218–227, 2006.

(UEGO) by its authors [3]. The separated hill-climbers work in restricted search regions (or clusters) of the search space. The volume of the clusters decreases as the search proceeds, resulting in a cooling effect similar to simulated annealing. Each cluster center represents diversity and quality, since it is result of hill-climbing procedures [3].

The scatter search (SS), proposed in [4], by another way, separates diversified and improved solutions in two sets: the reference set, containing the best solutions found so far and the diversity set, containing the solutions most distant from the solutions of the reference set. The solutions in these two sets are improved by local search. Thus, SS employs systematic exploration/exploitation moves, combining quality and representative solutions [4].

Clusters of mutually close solutions hopefully can correspond to relevant areas of attraction in the most of search metaheuristics, such as Genetic Algorithms (GA) [5] and Greedy Randomized Adaptive Search Procedure (GRASP) [6]. Relevant search areas can be treated with special interest by the algorithm as soon as they are discovered. This basic idea was first employed to propose the Evolutionary Clustering Search (ECS), applied to unconstrained continuous optimization [7]. Posteriorly, the search guided by clustering was extended to a GRASP with VNS (Variable Neighborhood Search[8]), and applied to Prize Collecting Traveling Salesman Problem (PCTSP) [9].

The clusters work as sliding windows, framing the search areas and giving a reference point (center) to problem-specific local search procedures. Furthermore, the cluster center itself is always updated by a permanent interaction with inner solutions, called assimilation [7,9].

This paper proposes the Clustering Search (*CS) as a generalized way of detecting promising search areas by clusters of solutions, suitable to be employed together with any metaheuristic and applicable to combinatorial and continuous optimization problems. To consolidate this approach as a flexible method, an ECS and a GRACS(Greedy Randomized Adaptive Clustering Search), both based on *CS, are proposed for pattern sequencing problems.

The remainder of this paper is organized as follows. In Section 2, the basic ideas and conceptual components of *CS are described. Theoretical issues of the sequencing problems are presented in Section 3. In section 4, the GRACS and the ECS are proposed for pattern sequencing problems. The computational results are examined in Section 5 and conclusions are summarized in Section 6.

2 Clustering Search Foundations

The *CS employs clustering for detecting promising areas of the search space. It is particularly interesting to find out such areas as soon as possible to change the search strategy over them. An area can be seen as a search subspace defined by a neighborhood relationship in metaheuristic coding space.

A cluster can be defined as a tuple $\mathcal{G} = \{c, r, s\}$, where c and r are the *center* and the *radius* of the area, respectively. The radius of a search area is the distance from its center to the edge. There also exist different *search strategies* s associated

to the clusters. Initially, the center c is obtained randomly and progressively it tends to slip along really promising points in the close subspace. The total cluster volume is defined by the radius r and can be calculated, considering the problem nature. It is important that r must define a search subspace suitable to be exploited by the search strategy s associated to the cluster.

For example, in unconstrained continuous optimization, it is possible to define r in a way that all search space is covered depending on the maximum number of clusters [7]. In combinatorial optimization, r can be defined as the number of movements needed to change a solution into another. In both case, the neighborhood is function of some distance metric related with the search strategy s, i.e., a problem-specific local search to be employed into the cluster.

2.1 Components

*CS can be splitted off in 4 conceptually independent parts: (a) a search metaheuristic (SM); (b) an iterative clustering (IC) component; (c) an analyzer module (AM); and (d) a local searcher (LS). Fig. 1 brings its conceptual design.

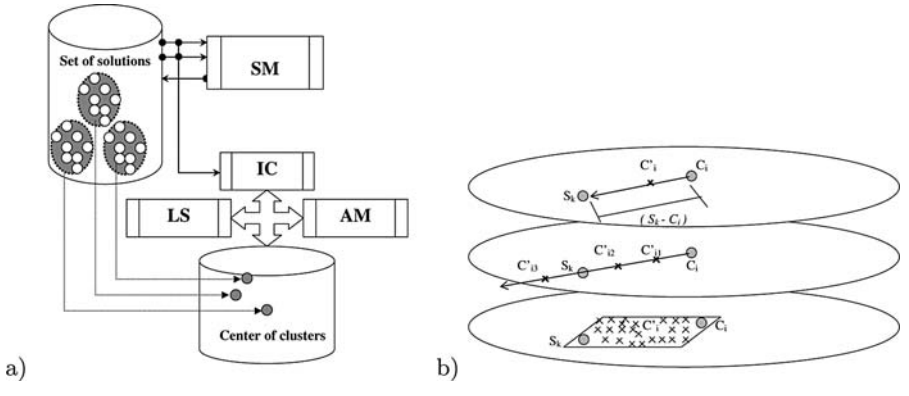


Fig. 1. a) *CS components; b) Simple, path and crossover assimilations, respectively

The SM component works as a full-time solution generator, according to its specific search strategy, performing independently of the remaining parts, and manipulating a set of $|P|$ solutions ($|P| > 1$ for evolutionary algorithms - EA). In an EA fashion, for example, individuals are selected, crossed over, and updated for the next generations. This entire process works like an infinite loop, in which solutions are generated along the iterations.

IC component aims to gather similar solutions into groups, maintaining a representative cluster center for them. To avoid extra computational effort, IC is designed as an online process, in which the clustering is progressively fed by solutions generated in each regular iteration of SM. A maximum number of clusters $\mathcal{NC}$ is a bound value that prevents a unlimited cluster creation. For a $n-$dimensional problem, the IC complexity is, at most, $O(\mathcal{NC} \cdot n)$ when all

cluster centers are allocated. A *distance metric*, $\wp$, must be defined, *a priori*, allowing a similarity measure for the clustering process.

AM component examines each cluster, in regular intervals, indicating a probable promising cluster. A *cluster density*, δ_i, is a measure that indicates the activity level inside the cluster i. For simplicity, δ_i counts the number of solutions generated by SM (selected solutions, in the EA case[7]). Whenever δ_i reaches a certain *threshold*, meaning that some information template becomes predominantly generated by SM, such information cluster must be better investigated to accelerate the convergence process on it. Clusters with lower δ_i are eliminated, as part of a mechanism that will allow creating other centers of information, keeping framed the most active of them. The cluster elimination does not affect the set of $|P|$ solutions in SM. Only the center of information is considered irrelevant for the process.

At last, the LS component is an internal searcher module that provides the exploitation of a supposed promising search area, framed by cluster. This process can happen after AM having discovered a target cluster or it can be a continuous process, inherent to IC, being performed whenever a new point is grouped. LS can be considered as the particular search strategy s associated with the cluster.

2.2 The Assimilation Process

Solutions generated by SM are passed to IC that attempts to group as known information, according to $\wp$. If the information is considered sufficiently new, it is kept as a center in a new cluster. Otherwise, redundant information activates a cluster, causing some kind of perturbation on it. This perturbation means an *assimilation process*, in which the previously learned knowledge (center of the cluster) is updated by the received information. More precisely, the assimilation process is applied over the closest center c_i, considering the new generated solution s_k. The general assimilation form is:

$$c_i' = c_i \oplus \beta(s_k \ominus c_i) \tag{1}$$

where $\oplus$ e $\ominus$ are abstract operations over c_i and s_k meaning, respectively, addition and subtraction of solutions. The operation $(s_k \ominus c_i)$ means the vector of differences between each one of the n variables compounding the solutions s_k and c_i, considering $\wp$. A certain percentage β of the vector is the update step for c_i, giving c_i'. According to β, the assimilation can assume different forms: simple, path and crossover assimilations, represented in Fig. 1b.

In simple assimilation, $\beta \in [0, 1]$ is a constant parameter, meaning a deterministic move of c_i in the direction of s_k. Only one internal point is generated more or less closer to c_i, depending on β, to be evaluated afterwards. The greater β, the less conservative the move is. This type of assimilation can be employed only with real-coded variables, where percentage of intervals is applicable. Its specific form is:

$$c_i' = c_i + \beta(s_k - c_i) \tag{2}$$

Although the name, crossover assimilation is not necessarily associated with an evolutionary operator. In a general way, it means any random operation between two candidate solutions, giving other ones, similarly as a crossover operation in EAs. In this assimilation, β is an $n-$dimensional random vector and c'_i can assume a random point inside the hyper plane containing s_k e c_i. Since the whole operation is a crossover or other binary operator between s_k and c_i, it can be applied to any type of coding or even problem (combinatorial or continuous one). The $\overrightarrow{\beta}$ parameter is resulting from the type of crossover employed, not the crossover parameter itself. The crossover assimilation can be written by:

$$c'_i = c_i + \overrightarrow{\beta} \cdot (s_k - c_i) \tag{3}$$

Simple and crossover assimilations generate only one internal point to be evaluated afterwards. Path assimilation, instead, can generate several internal points or even external ones, holding the best evaluated one to be the new center. It seems to be advantageous, but clearly costly. These exploratory moves are commonly referred in path relinking theory [10]. In path assimilation, β is a $\eta-$dimensional vector of constant and evenly spaced parameters, used to generate η samples taken in the path connecting c_i and s_k. Since each sample is evaluated by the objective function, the path assimilation itself is an intensification mechanism inside the clusters. The new center c'_i is given by:

$$\begin{array}{rl} c'_i = c'_V,\, f(c'_V) = \min\left\{f(c'_1), f(c'_2), \cdots, f(c'_\eta)\right\} & \\ c'_j \quad = c_i + \beta_j(s_k - c_i) & \\ \beta_j \quad \in \{\beta_1, \beta_2, \cdots, \beta_\eta\} & \end{array} \tag{4}$$

where $\beta_j \in \{]0, 1[\bigcup]1, \infty]\}$, $f(c'_V)$ is the objective function of the best evaluated solution sampled in the path and min is concerned to minimization problems.

With respect to the infinite interval in (4), it means the external points can be sampled indefinitely while there are well-succeeded points beyond s_k. A well-succeeded point has an objective function value better than the previous point sampled, in a way that a worse point stops the sampling. In the Fig. 1b, the point c_{i3} is evaluated after s_k. Such extrapolation move is suitable for path relinking [10] and it can intentionally shift the center cluster beyond the cluster edge.

3 Theoretical Issues of the Pattern Sequencing Problem

Pattern sequencing problems may be stated by a matrix with integer elements where the objective is to find a permutation (or sequencing) of rows or patterns (client orders, or gates in a VLSI circuit, or cutting patterns) minimizing some objective function [11]. Objective functions considered here differ from traveling salesman-like problems because the evaluation of a permutation can not be computed by using values that only depend on adjacent patterns. There are two similar pattern sequencing problems found in the literature: Minimization of Open Stacks Problem (MOSP) and Gate Matrix Layout Problem (GMLP) [12]. The difference between them resides only in their enunciation. This work is considering only the largest GMLP instances found in the literature.

GMLPs are related to one-dimensional logic arrays and programmable logic arrays folding [13]. The data for a GMLP are given by an $I \times J$ binary matrix $\mathbf{P}$, representing gates (rows) and nets (columns), where $\mathbf{P}_{ij} = 1$, if gate i belongs to net j, and $\mathbf{P}_{ij} = 0$ otherwise. Non-overlapping nets can be placed at the same connection track, minimizing the layout area (cost). A GMLP consists of determining a sequence of gate that minimizes the maximum of tracks (MOT) in the circuit. Another binary matrix, here called matrix $\mathbf{Q}$, can be used to calculate the MOT for a certain gate permutation. It is derived from the input matrix $\mathbf{P}$, by the following rules:

- $\mathbf{Q}_{ij} = 1$ if there exists x and $y | \pi(x) \leq i \leq \pi(y)$ and $\mathbf{P}_{xj} = \mathbf{P}_{yj} = 1$;
- $\mathbf{Q}_{ij} = 0$, otherwise;

where $\pi(b)$ is the position of gate b in the permutation. The $\mathbf{Q}$ shows the consecutive-ones property [14] applied to $\mathbf{P}$: in each column, "0" 's between "1" 's are replaced by "1" 's. The sum of "1" 's, by row, computes the number of open stacks when each pattern is processed. Fig. 2 shows an example of matrix $\mathbf{P}$, its corresponding matrix $\mathbf{Q}$, and the number of tracks in the circuit. At most, 7 tracks ($MOT = \max\{3, 3, 3, 5, 6, 7, 7, 5, 3\} = 7$) are needed to manufacture a circuit with permutation $\pi_0 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

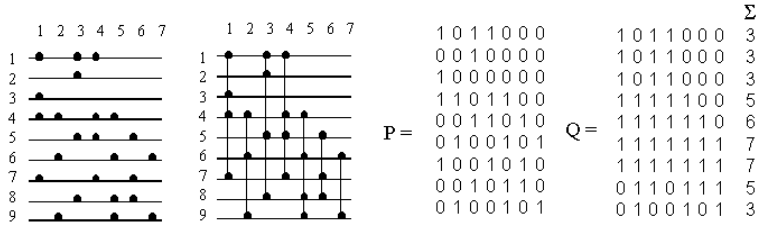


Fig. 2. GMLP (or MOSP) instance: matrix $\mathbf{P}$ and corresponding $\mathbf{Q}$

4 GRACS and ECS Implementations

Two *CS metaheuristics are now described: a GRACS (Greedy Randomized Adaptive Clustering Search) and an Evolutionary Clustering Search ECS. In the latter, the component SM is a steady-state GA employing well-known genetic operators as roulette wheel selection [5], block-order crossover (BOX) [15], and 2-swap mutation [16]. In BOX, the parent(A) and parent(B) are mixed into only one offspring, by copying blocks of both parents, at random. Pieces copied from a parent are not copied from other, keeping the offspring feasible (Fig. 3a).

In general, GRASP consists of a greedy construction phase and a subsequent one which iterative local search improvements are made in the previously obtained greedy solution. In GRACS, the component SM is a modified GRASP, removing the native local search procedure from it and adding the IC-AM) components, which are responsible for intermediating the local search calls. Instead

of always to perform the local search, the improvements are made according to the IC-AM criterion for promising search areas.

The constructive greedy procedure chosen for pattern sequencing problems is based on the rule for filling schemata [17], which says that the new pattern to be included in greedy solution shall minimize a bit-to-bit *xor* difference with respect to the previous included. In other words, examining all the J columns (nets) of the candidate list, matching bit-to-bit and summing all the xor results ($xor(1,0) = xor(0,1) = 1$ and $xor(0,0) = xor(1,1) = 0$), the patterns are included in a sequence that improves the similarity between adjacent ones. A typical GRASP parameter, α, that balances the greedy/random behaviour of the constructive phase, was set to 0.10, meaning that only 10% of the candidate list is considered to choose the next included pattern in solution.

The component LS in both *CS algorithms was implemented by a 2-Opt hill-climbing procedure which is applied to the center of promising cluster. The hill-climbing explores a search tree, considering several 2-Opt neighborhoods (Fig. 3b). The best neighbor from a level (bold circle) is taken as starting point to the next, respecting a maximum width l (maximum number of swaps at each level) and height m (maximum number of levels). For GRACS, m was set unlimited, i.e., while better solutions were being found. Rather, for ECS, $m \leq 40$. For both, ECS and GRACS, l was set to 0.70, meaning that 70% of the patterns can be exchanged in each level during the local search procedure.

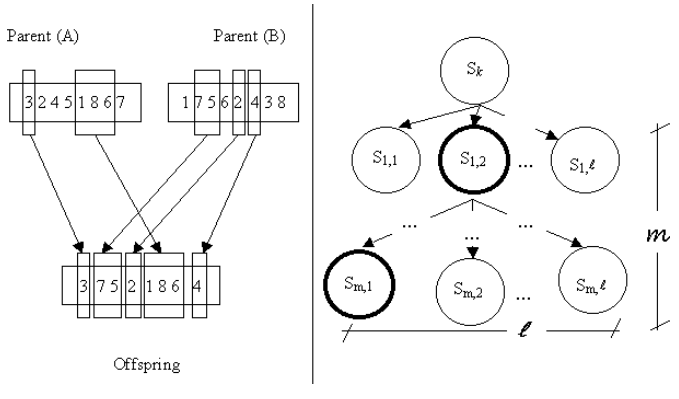


Fig. 3. (a) Block order crossover and (b) 2-Opt hill-climbing tree

Concerning to component IC, for both algorithms, the 2-swap distance metric is employed, i.e., the number of 2-swap needed to move a solution, along the search space, to another. Identical solutions need no changes to turn one into other. By the other side, completely distinct solutions may need about $I - 1$ 2-swap moves to lead a point to another. The radius of a cluster is given by:

$$r_t = \lceil 0,9I \rceil \tag{5}$$

Table 1. Example of full path between center c_i and new point s_k

c_i=	1 2 3 4 5 6 7 8 9	comparison	swap	evaluation
1)	4 2 3 1 5 6 7 8 9	1	1	1
2)	4 8 3 1 5 6 7 2 9	1	1	1
3)	4 8 5 1 3 6 7 2 9	1	1	1
4)	4 8 5 9 3 6 7 2 1	1	1	1
5)	4 8 5 9 1 6 7 2 3	1	1	1
6)	4 8 5 9 1 7 6 2 3	1	1	1
7)	4 8 5 9 1 7 6 2 3	1		
8)	4 8 5 9 1 7 6 2 3	1		
s_k=	4 8 5 9 1 7 6 2 3	8	6	6

i.e., a relatively greater radius, because it requires only 10% of labels matching for a given pattern sequencing to be considered close enough to the center of a cluster. Whenever a selected individual s_k is *far away* from all centers (a distance above r_t), then a new cluster must be created.

In this application, the path assimilation was chosen. The more distance $\wp(c_i, s_k)$, the more potential solutions exist between c_i and s_k. The sampling process, depending on the number of instance variables, can be costly, since each solution must be evaluated by objective function. In Table 1, a completely 2-swap path between two solutions, c_i and s_k, can be seen.

Each comparison means one iteration in the assimilation algorithm which also can occur one swap/evaluation of the intermediary points. At last, the center will be shiffted to the best point evaluated in this path. Actually, there have been occurred 6 pattern swaps and, consequently, 6 objective function calls. The distance $\wp(c_i, s_k)$ is not necessarily 6 because other paths with distance less than 6 could be found. However, **CS* applications require computing such distance to associate the point to a particular center during the clustering process. Therefore, $\wp(c_i, s_k)$ is estimate considering the number of pattern in different positions in each permutation (variables that do not match). This value is still decremented by one, because even all I patterns were in different positions in each permutation, it would be generated at most $I - 1$ intermediary solutions.

5 Computational Results

ECS, GRACS and GRASP were coded in ANSI C and were run on Intel AMD (1.33 GHz) platform. The most important performance parameters were set as follows: for ECS, $20 \leq \mathcal{NC} \leq 30$ and $300 \leq |P| \leq 500$; for GRACS and GRASP, $\alpha = 0.10$ and $l = 0.7$. For each instance, were performed 20 trials, allowing the approaches to perform a maximum number of objective function calls. These parameter values try to make the tuning for the algorithm speed-accuracy trade-off and they were chosen through the authors' expertise.

ECS, GRACS and GRASP are now compared against the Parallel Memetic Algorithm (PMA)[16]. Besides a parallel algorithm, employing a suitable migration

policy, PMA presents a new 2-swap local search with a reduction scheme, which discards useless swaps, avoiding unnecessary objective function calls. Its results were considered so far the best ones obtained in the literature, specifically with large GMLP instances [16].

The Table 2 shows the comparison between all *CS approaches and PMA. For the latter, the results were obtained in 10 trials[16]. GRASP was included for verifying a probable improvement by the clustering process, since GRACS is a modified GRASP. The success rate (SR) to reach the best known solution as well as the average of the number of objective function calls (FC) were considered for measuring the algorithm performances. For each tested instance, the result in bold shows the winning approach. In 3 of 5 instances, at least one of the *CS approaches was better than PMA. But in the particular contest between GRACS and GRASP, both performances were very similar. In $w4$ instance, GRASP has obtained better SR, but requiring more FCs. This fact evidences the need of better tuning GRACS, since GRASP/GRACS appear to be very promising approaches, with SR comparable with the best results found in the literature. For instance $w4$, the largest one found in literature, ECS has reached meaningful best results.

Table 2. Results of ECS, GRACS, GRASP and PMA for GMLP instances

Inst.(IxJ)	ECS		GRACS		GRASP		PMA	
	SR(%)	FC	SR(%)	FC	SR(%)	FC	SR(%)	FC
x0 (48x40)	100	119296.0	100	39187.8	**100**	**24662.6**	100	43033.0
v4470 (47x37)	60	169136.0	**100**	**90459.8**	100	98081.0	60	176631.0
w2 (33x48)	100	26185.0	100	10002.6	100	12580.7	**100**	**3523.0**
w3 (70x84)	50	540893.0	90	372021.0	90	360225.4	**90**	**203892.0**
w4 (141x202)	**55**	**1695924.0**	20	2357496.0	30	3998868.0	20	9428591.0

6 Conclusion

This paper proposes a new way of detecting promising search areas based on clustering: the Clustering Search (*CS). Together with other search metaheuristics, working as full-time solution generators, *CS attempts to locate promising search areas by solution clustering. The clusters work as sliding windows, framing the search areas and giving a reference point to problem-specific local search procedures, besides an iterative process, called assimilation.

Two metaheuristics based on *CS were also proposed for large scale GMLP instances: a GRACS (Greedy Randomized Adaptive Clustering Search) and an Evolutionary Clustering Search ECS. In comparison against the best results found in the literature, *CS approaches have achieved similar and sometimes superior performance. However, in the particular contest between GRACS and GRASP, both performances were very similar, evidencing the need of further tuning GRACS. Besides, to combine clustering with other metaheuristics as Immune Systems and Evolution Strategies will be considered in further research.

References

1. Moscato, P.: Memetic algorithms: a short introduction. In: Corne, D.; Dorigo, M.; Glover, F. (eds) New Ideas in Optimization. London: McGraw-Hill (1999) 219-234
2. Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-coded memetic algorithms with crossover hill-climbing, Evol. Computation (2004) 12(3):273-302
3. Jelasity, M., Ortigosa, P., García, I.: UEGO, an Abstract Clustering Technique for Multimodal Global Optimization, Journal of Heuristics (2001) 7(3):215-233
4. Glover, F.: A template for scatter search and path relinking. Selected Papers from the Third European Conference on Artificial Evolution. Springer-Verlag (1998) 3-54
5. Goldberg, D.E.: Genetic algorithms in search, optimisation and machine learning. Addison-Wesley (1989)
6. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. Journal of Global Optimization (1995) 6:109-133
7. Oliveira, A.C.M, Lorena, L.A.N.: Detecting promising areas by evolutionary clustering search. In: SBIA 2004, 17, (LNAI) v. 3171 (2004) 385-394
8. Mladenovic, N., Hansen, P.: Variable neighborhood search. Computers and Operations Research(1997) 24:1097-1100
9. Chaves, A.A., Lorena, L.A.N.: Hybrid algorithms with detection of promising areas for the prize collecting travelling salesman problem. In: HIS 2005, 5 (2005) 49-54
10. Glover, F., Laguna, M., Martí, R.: Fundamentals of scatter search and path relinking. Control and Cybernetics(2000) 39:653-684
11. Fink, A., Voss, S.: Applications of modern heuristic search methods to pattern sequencing problems, Computers and Operations Research (1999) 26(1):17-34
12. Linhares, A.: Industrial pattern sequencing problems: some complexity results and new local search models. Doctoral Thesis, INPE, S. José dos Campos, Brazil (2002)
13. Möhring, R.: Graph problems related to gate matrix layout and PLA folding, *Computing* (1990) 7:17-51
14. Golumbic, M.: Algorithmic graph theory and perfect graphs. *Academic Press*, New York (1980)
15. Syswerda, G.: Schedule optimization using genetic algorithms. *Handbook of Genetic Algorithms*,Van Nostrand Reinhold, New York (1991) 332-349
16. Mendes, A., Linhares, A.: A multiple population evolutionary approach to gate matrix layout, Systems Science, Taylor & Francis (2004) 35(1):13-23
17. Oliveira A.C.M., Lorena, L.A.N.: A Constructive Genetic Algorithm for Gate Matrix Layout Problems. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (2002) 21(8):969-974

Development of a Hybrid Intelligent System for Electrical Load Forecasting

Ronaldo R.B. de Aquino[1], Aida A. Ferreira[2], Manoel A. Carvalho Jr.[1], Milde M.S. Lira[1], Geane B. Silva[1], and Otoni Nóbrega Neto[1]

[1] Federal University of Pernambuco, Academico Helio Ramos s/n, Cidade Universitária, Cep: 50.740-530 – Recife – PE – Brazil
rrba@ufpe.br, macj@ufpe.br, milde@ufpe.br, geaneufpe@yahoo.com.br, otoninobrega@hotmail.com
[2] Federal Center of Technologic Education of Pernambuco, Av Professor Luis Freire, 500, Cidade Universitária, Cep:50.740-530 – Recife – PE – Brazil
aidaaf@gmail.com

Abstract. This paper presents a hybrid intelligent system for electrical load forecast. Artificial Neural Networks (ANN) were combined with Heuristic Rules to create the system. The study was based on load demand data of Energy Company of Pernambuco (CELPE), whose data contain the hourly load consumption in the period from January-2000 until December-2004. The data of hourly consumption of the holidays were eliminated from the file, as well as the data regarding the more critical period of the rationing in Brazil (from May until July 2001). The hybrid intelligent system presented an improvement in the load forecasts in relation to the results achieved by the ANN alone. The system was implemented in MATLAB.

Keywords: Artificial Neural Networks (ANN), Hybrid System, Heuristic Rules, Electric Load Forecast.

1 Introduction

The daily operation and planning activities of an electric utility requires the prediction of the electrical demand of its customers. Several researches have been carried out in order to improve planning and operation of these systems. Specifically, the required load forecasts may be divided into short-term, mid-term and long-term forecasts.

Traditionally, load forecasts techniques use statistical methods of time series analysis, which include linear regression, exponential damping and Box Jenkins [1]. In recent years, techniques of artificial intelligence such as Artificial Neural Networks (ANN) have been used, obtaining promising results [2]-[6].

Currently, the procedure adopted by CELPE for hourly load forecasts is a mixing of statistical techniques with specialists´ knowledge. The aim of this work is to improve the hourly load forecast, automating it and incorporating the implicit knowledge of the specialist. The developed system (named PREVER and implemented in MATLAB®) makes use of a hybrid approach of ANN based techniques and Heuristic Rules to adjust the short and mid-term electric load forecasting in 7 and 45 days ahead.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 228 – 237, 2006.

2 Creating the Data Base

The data used in this work were made available by CELPE and they correspond to the hourly load demand data in the period from January 2000 until December 2004. The problem approached in this work is based on the hourly load forecasting in 7 and 45 days ahead.

All the data were unified in a single file, where each pattern was arranged by the information of the year, day, month and the load of the day for every hour (24 hours) and the day of the week to be forecast (Sunday, Monday,.... Saturday). The data regarding the more critical period of the rationing (from May to July of 2001) were eliminated from the file. The hourly load data were normalized (LN) to fall in the range 0 to 1 by using (1):

$$L_N = \frac{L - L_{min}}{L_{max} - L_{min}} \ . \tag{1}$$

where L_N is the hourly load value registered by the CELPE's system, L_{max} and L_{min} are the maximum and the minimum hourly load value among all the observed values, respectively. In this work $L_{min} = 0$ and $L_{max} = 1.1 \cdot L_{A\_max}$, where L_{A_max} is the maximum value of the actual load data. The objective of factor 1.1 is to turn the values of future loads up to 10% above L_{A_max} into values below the unit after their normalization.

In this work, a holiday is considered by the specialist as one Saturday or one Sunday, according to [7]. In other words, the specialist indicates if the load behavior of that specific holiday is more correlated with the load behavior of Saturday or Sunday. Because the load curves of the holidays are close to the load curves of one Saturday or one Sunday, the hourly load data of holidays were just used in the test set. The training of Multilayer Perceptron (MLP) networks follows a paradigm of supervised learning, where each pattern in the training set is represented by an input and a desired output pairs. The patterns of the input set have the following arrangement: The first 24 values correspond to the hourly consumption of (***n+1***) days before the day to be forecast, the next 24 values correspond to the hourly consumption of ***n*** days before the day to be forecast (***n*** = 7 or 45 days), and finally, the next 7 values define the day of the week that will be forecast (Sunday, Monday,.... Saturday). This information used *1-of-m* code.

The data base, formed by 969 examples of each days ahead, is distributed in the following way: 60% for the training set, 30% for the validation set and 10% for the test set. The patterns of each group were selected in a random way.

The main objective of the load forecasting system based on ANN is to learn from pattern of known values and to generalize for new ones. The performance of the system will be measured by percentage of the mean-square error (MSE) [8] specified in (2), and by the mean absolute percentage error (MAPE) in (3).

$$MSE_{\%} = 100 \times \frac{L_{max} - L_{mín}}{N \cdot P} \sum_{p=1}^{P} \sum_{i=1}^{N} (L_{pi} - T_{pi})^2 \ . \tag{2}$$

where $L_{\max}$ and $L_{\min}$ are the maximum and minimum of the hourly load values, in the representation of the problem, respectively; N is the number of output units of the ANN; P is the total number of patterns in data base; L_{pi} and T_{pi} are actual and desired target output of the i^{th} neuron in the output layer, respectively.

$$MAPE_{\%} = \frac{1}{P}\sum_{p=1}^{P}\frac{\left|L_p - T_p\right|}{T_p} \times 100 \cdot \quad (3)$$

where P is the total number of patterns in data base; L_p and T_p are the actual and desired output value for a given input, respectively.

The ANNs involved are designed using the method of training-and-testing. The basic idea of this method is to divide the number of pattern in three mutually exclusive subsets. The first subset is the training set used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process to avoid *overfitting*. The third subset is the test set used exclusively for measuring the error of the system. The idea is that the performance of the system in a test set is its performance in the same task of the real world. This means that no information of the test set can be available during the training [8].

Attempting to achieve an estimate error nearest to the true error, the 10-fold cross validation method was chosen to generate the training, validation and test sets. This method has become a standard method in practical terms [9], [10]. Therefore the patterns were divided in ten independent partitions, and each partition has 10% of the data. In each experiment three partitions were used to validate, one, to test and the six remaining partitions were used to train the ANN. Table 1 presents the division of the partitions.

Table 1. The Sets Arrangements

Experiments	Neural Network	Training	Validation	Test
1	1 – 10	5, 6, 7, 8, 9, 10	2, 3, 4	1
2	11 - 20	1, 6, 7, 8, 9, 10	3, 4, 5	2
3	21 - 30	1, 2, 7, 8, 9, 10	4, 5, 6,	3
4	31 - 40	1, 2, 3, 8, 9, 10	5, 6, 7	4
5	41 - 50	1, 2, 3, 4, 9, 10	6, 7, 8	5
6	51 - 60	1, 2, 3, 4, 5, 10	7, 8, 9	6
7	61 - 70	1, 2, 3, 4, 5, 6	8, 9, 10	7
8	71 - 80	2, 3, 4, 5, 6, 7	9, 10, 1	8
9	81 - 90	3, 4, 5, 6, 7, 8	1, 2, 10	9
10	91 – 100	4, 5, 6, 7, 8, 9	1, 2, 3	10

3 Artificial Neural Networks

All of the ANN used were created with MLP architecture, using resilient backpropagation (RPROP) training algorithm [11]. They have an input layer, a hidden layer and

an output layer. The nodes of the hidden layer use the tan-sigmoid activation function and those of the output layer use the log-sigmoid activation function. The maximum number of iterations for all of the trainings was set to 2500 epochs. The training stops if the early stopping implemented by MATLAB® happens 20 times consecutively, or if the maximum number of epochs is reached, or if the error gradient reaches a minimum, or still if the error goal in the training set is met. The early stopping method has the objective of improving generalization of the neural networks. MATLAB® implements this technique, monitoring the error on the validation set during the training process.

All of the neural networks developed have 55 nodes in the input layer distributed in the following way: 24 nodes correspond to the hourly consumption of ***(n+1)*** days before the day to be forecasted, 24 nodes correspond to the hourly consumption of ***n*** days before the day to be forecasted (***n*** = 7 or 45 days), and finally, the next 7 values define the day of the week that will be forecast using *1-of-m* code (e.g. Sunday='1000000'). The output layer is characterized by 24 values, one for each hour of the day, which indicates the hourly load consumption of the day to be forecasted.

To decide for the best configuration of nodes in the hidden layer in several horizons, ten experiments were carried out with random initialization of weights and with varying number of hidden nodes from 30 to 130 with an increment of 5. The number of hidden nodes in the best neural networks for the respective horizons is presented in Table 2.

Table 2. Number of hidden nodes

Days ahead	7	45
Hidden Nodes	95	100

4 Improving the Forecast

In this paper, these rules have been developed with the specific aim of reducing the error of the hourly load forecasting accomplished by the neural network. In the area of expert system design, representations of heuristic rules have been extensively studied [12]. Two types of heuristic rules have been developed. In the first one (horizons of 7 days ahead), the adjustment of the neural network output is made using the average and the standard deviation of the hourly historical load consumption. In the second one (horizons 45 days ahead), the adjustment is accomplished by the average and the monthly historical load consumption.

Rule 1: Short-Term Load Forecast

This heuristic rule is used to adjust the ANN hourly load forecasting output in the horizons of 7 days ahead. The adjustment is accomplish by evaluating the average of the consumption of the last 3 days which have the same characteristic as the prediction day and whose date are lower or equal to the difference between the date of the prediction day and the horizon. For instance, let the date of the prediction day be

04/24/2005 (Sunday) in the horizon of 7 days. The average is computed in the following way: taking the prediction day minus the horizon gives 04/17/2005 (Sunday). As the prediction day is a Sunday, 3 previous consecutive Sundays should be taken before the date 04/17/2005. Thus, the average will be computed using the consumption of the days 04/10/2005, 04/03/2005 and 03/27/2005.

For the case of usual day, we take the consumption of the days that have approximately the same load curve of the prediction day. That is, if we want to forecast a Monday, the average will be computed by the consumption of previous Mondays. For the case of holiday, the average will be calculated using Sunday or Saturday, which depends on how the specific holiday was registered in the system by the specialist.

After the calculation of the average hourly consumption, we can compute the standard deviation. In the next step, we compute the upper and lower limits of the confidence interval, where the upper limit is the average consumption plus the standard deviation and the lower limit is the average consumption minus the standard deviation. If the value of the neural network output is out of the confidence interval, its output is adjusted by the average, otherwise, it remains unaffected. This procedure makes the hourly load forecasting by the neural network to fall in the confidence interval.

Rule 2: Mid-Term Load Forecast

This heuristic rule is used to adjust the ANN hourly load forecasting output for 45 days ahead. Here again the consumption average was considered, and beside this, the consumption increase or decrease from one month to the next according to the historical seasonality.

Due to the lack of the month of the prediction day in the neural network input, it was necessary to add this information to create this heuristic rule, which was made by an increase or a decrease factor.

To find the increase or decrease factor, the daily load mean consumption was calculated, dividing the monthly consumption by the day's number of the month. Next the load behavior of one month to the next was analyzed. The factors were computed as a function of the horizon and can be summarized as follows:

The monthly factor for 45 days ahead is the average of the factors between the monthly factors of 30 and 60 days. The factor for 30 days ahead was calculated dividing the mean daily consumption from the current month by the previous one, reducing this result by the unit. The factor for 60 days ahead was calculated dividing the mean daily consumption from the current month by the penultimate month, reducing this result by the unit.

If the final result is positive, it means that there was an increase in the consumption of the previous month in comparison with current month, otherwise, there was a reduction. In the next step, the single monthly factor which corresponds to the average of the factor calculated previously in the periods from January 1994 until December 2004 was calculated. Table 3 shows the monthly adjusted factor for 45 days ahead.

After computing the factor, an algorithm was achieved so that the network output improves the hourly load forecasting. The rules are then used to adjust the output of the network.

The rules are based on the comparison between the hourly load forecasting by the neural network and a reference value. This reference value is given by the average of the last consumptions added to the portion of the monthly behavior, which corresponds to the multiplication of the monthly factor and the average of the consumptions. This average is the same average described in Rule 1, which is applied to 7 days ahead.

Table 3. Monthly Adjusted Factor

Month	45 days
January	0.00236
February	0.00544
March	0.01495
April	-0.00320
May	-0.01730
June	-0.04430
July	-0.03320
August	0.01252
September	0.05576
October	0.05790
November	0.04227
December	0.02167

Finally, the Rules can be stated:

Positive monthly factor: If the simulated value by the neural network is smaller than the reference value, the neural network output will be adjusted to the reference value. Otherwise, it remains unaffected;

Negative monthly factor: If the simulated value by the neural network is larger than the reference value, the neural network output will be adjusted to the reference value. Otherwise, it remains unaffected;

Specific Rules for January first and second: January 1, the first day of the New Year has the smallest load consumption of the year, in other words, it is a pattern different from the others kind of patterns presented during the training process. On the other hand, January 2 suffers the consequence of January 1, and presents a low consumption in relation to the other days of January. So a specific rule for these days had to be made. If the system is forecasting the consumption on January first or second, it should obey the following rule:

- If the hour of the day is between 1 and 6, the adjusted value should be multiplied by 0.92;
- If the hour of the day is between 6 and 20, the adjusted value should be multiplied by 0.94;
- If the hour of the day is between 20 and 24, the adjusted value should be multiplied by 0.97.

It is important to point out that the adjusted value be the value after the adjustment is accomplished according to the forecast horizon. For instance, if these days are being forecasted in 7 days ahead, the adjusted value will be the value of the network output after the adjustment in accordance with rule 1.

The system is implemented in the way that a user can accomplish his/her own adjustments manually for each day or hours of the day to be forecasted, similar to those made for January 1 and 2.

5 Results

In order to verify the performance of the forecasting system with ANN and the forecasting with the Hybrid System (ANN + Heuristics Rules) on the load data pertaining to CELPE, forecasts were accomplished, short and mid-term, in the period from January until October 2005. Table 4 shows the mean hourly MAPE for 7 days ahead with ANN and the Hybrid System.

Table 4. Mean hourly MAPE for 7 days ahead

Month	ANN Mape	Hybrid System Mape
January	3.80	2.21
February	3.36	2.54
March	4.33	3.99
April	3.40	2.50
May	2.59	4.22
June	2.76	2.73
July	2.14	1.87
August	2.53	2.13
September	3.45	3.35
October	2.98	2.46
Mean	**3.13**	**2.80**

In the period of 10 months from January to October 2005, only in May was the ANN superior to Hybrid System for 7 days load forecast. Moreover, the mean hourly MAPE in each month for the ANN in the period was *3.13* as against *2.80* presented by the Hybrid System. Thus, the hybrid system for electrical load forecast 7 days ahead is certainly an improvement on the ANN forecast. The results obtained in references [2] and [3] for 7 days ahead load forecast are similar to that got in this paper.

However, just the results of the 45 days ahead will be compared here, because CELPE distribution utility carries out its load forecasting just in this period. This permit making a comparison of the Hybrid System and the forecasting model currently being used by CELPE. In Fig. 1, there are the graphs of the real consumption, of the load forecasting system performed by CELPE, ANN and Hybrid System (ANN plus adjustment) with 45 days ahead.

After examining the curves of the normalized monthly consumption (Fig. 1), it can be seeing that the curve of ANN plus adjustment is the closest to the curve of the real consumption. This fact demonstrates the superiority of the hybrid system.

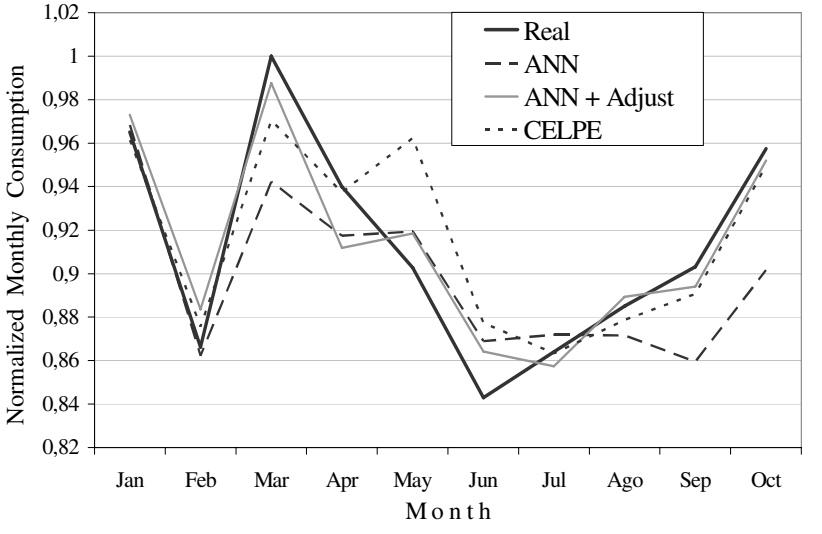


Fig. 1. Normalized Monthly Consumption

Table 5. Mean hourly MAPE and MSE

	ANN		**Hybrid System**		**CELPE**	
Month	Mape	MSE	Mape	MSE	Mape	MSE
January	3.09	0.07	2.55	0.06	3.32	0.10
February	3.43	0.09	3.23	0.08	3.28	0.09
March	6.13	0.26	3.45	0.09	5.13	0.20
April	3.74	0.10	3.75	0.11	2.96	0.07
May	2.95	0.06	2.89	0.05	7.32	0.25
June	4.04	0.10	3.55	0.08	5.24	0.14
July	2.18	0.03	2.32	0.03	2.72	0.05
August	3.15	0.06	2.55	0.04	2.98	0.05
September	4.90	0.15	2.52	0.04	3.16	0.07
October	5.90	0.22	2.27	0.04	2.35	0.04
Mean	**3.95**	**0.11**	**2.91**	**0.06**	**3.85**	**0.11**

Table 5 shows the mean hourly MAPE and the MSE for the same systems. In the period of 10 months from January to October 2005, only in April was the system of

CELPE superior to Hybrid System. Moreover, the mean hourly MAPE in each month for the Hybrid System in the period was *2.91* as against *3.85* presented by the CELPE system. The mean hourly MSE in each month for the PREVER system in ten months was *0.06* as against *0.11* presented by the CELPE system. Thus, the Hybrid System is certainly an improvement on the CELPE system.

6 Conclusion

This work presents the results of short and mid-term load forecasting by a Hybrid System. Applying a hybrid intelligent system approach of ANN based technique and heuristic rules, this system is able to forecast the electric load of CELPE system.

The Hybrid System was evaluated in all forecast horizons mentioned previously. However, only the 45 days ahead was compared here because the CELPE´s load forecast is only carried out in this period for making their comparisons possible. The results confirmed the potential and suitability of the hybrid intelligent system implemented compared to CELPE´s load forecasting system. In the period from January until October 2005, Hybrid System was more precise than CELPE´s load forecasting system over 10 months.

The results obtained here for 7 days ahead load forecast are similar with reference [2] and [3].

References

1. D. C. Montgomery, L. A. Johnson, and J. S. Gardiner, “Forecasting and Time Series Analysis.” McGraw-Hill International Editions, 1990.
2. A. Bakirtzis, V. Petrldis, S. J. Kiartiz, and M. C. Alexiardis, “A Neural Network Short Term Load Forecasting Model for the Greek Power System,” IEEE Transactions on Power Systems, vol.11, no. 2, 1996, pp 858-863.
3. A. Khotanzad, R. Afkhami-rohani, D. J. Maratukulam, “ANNSTLF - Artificial Neural Network Short-Term Load Forecaster-Generation Three” IEEE Transactions on Power Systems, Vol. 13, No. 4, pp. 1413-1422, November,1998.
4. C. Kim, I. Yu, and Y. H. Song, “Kohonen Neural Network and Wavelet Transform Based Approach to Short-Term Load Forecasting,” Electric Power Systems Research, vol. 63, issue 3, 2002, pp. 169-176.
5. S. E. Papadakis, J. B. Theocharis, S. J. Kiartzis, and A. G. Bakirtzis, “A novel approach to Short-term Load Forecasting using Fuzzy neural networks,” IEEE Transactions on Power Systems, vol. 13, no. 2, 1998, pp. 480-492.
6. A. Silva, L. Moulin, and A. J. R. Reis, “Feature Extraction via Multiresolution Analysis for Short-Term Load Forecasting,” IEEE Transactions on Power Systems, vol. 20, no. 1, 2005, pp. 189-198.
7. R. R. B. Aquino, A. A. Ferreira, G. B. Silva, O. Nóbrega Neto, M. M. S. Lira, and J. B. Oliveira “Previsão de Carga Horária em Médio Prazo Utilizando Redes Neurais com Foco na Previsão dos Feriados,” VII CBRN – Congresso Brasileiro de Redes Neurais, ISSN: 1808-8589, Natal-RN, Brazil, 2005 (in Portuguese).
8. L. Prechelt, “Proben1 – A Set of Neural Network Benchmark Problems and Benchmarking Rules,” Technical Report, 1994, pp. 21-94.

9. A. A. Ferreira, "Comparação de arquiteturas de redes neurais para sistemas de reconhecimento de padrões em narizes artificiais," Master dissertation, Center of Informatics, Federal University of Pernambuco, Recife, Brazil, 2004 (in Portuguese).
10. W. Witten and E. Frank, "Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations," Morgan Kaufmann Publishers, 2000, pp. 126.
11. M. Riedmiller and H. Braun, "A Direct adaptive Method for faster Backpropagation Learning: The RPROP Algorithm," IEEE International Conference on Neural Networks, vol.1, April 1993, pp.586-591.
12. A. J. Gonzales and D. D. Dankel, "The Engineering of Knowledge-Based Systems," Prentice Hall, 1993.

Extending a Hybrid CBR-ANN Model by Modeling Predictive Attributes Using Fuzzy Sets*

Yanet Rodriguez[1], Maria M. Garcia[1], Bernard De Baets[2],
Rafael Bello[1], and Carlos Morell[1]

[1] Universidad Central de Las Villas, Carretera a Camajuani km 51/2, Santa Clara, Cuba.
{yrsarabia, mmgarcia, rbellop, cmorellp}@uclv.edu.cu
[2] Ghent University, Coupure links 653, B-9000 Gent, Belgium
Bernard.DeBaets@UGent.be

Abstract. This paper presents an extension of an existing hybrid model for the development of knowledge-based systems, combining case-based reasoning (CBR) and artificial neural networks (ANN). The extension consists of the modeling of predictive attributes in terms of fuzzy sets. As such, representative values for numeric attributes are fuzzy sets, facilitating the use of natural language, thus accounting for words with ambiguous meanings. The topology and learning of the associative ANN are based on these representative values. The ANN is used for suggesting the value of the target attribute for a given query. Afterwards, the case-based module justifies the solution provided by the ANN using a similarity function, which includes the weights of the ANN and the membership degrees in the fuzzy sets considered. Experimental results show that the proposed model preserves the advantages of the hybridization used in the original model, while guaranteeing robustness and interpretability.

1 Introduction

One of the most important characteristics of Knowledge-Based Systems (KBS) is their capability to explain the obtained results. The most common way to do this is by using a rule-based system in which the relationship of the rules leading to the reached conclusion is presented [1]. Another way to explain the results is by "justification", where relevant information in achieving a result is presented [2]. Kolodner in [3] defines the Interpretative Case-based Reasoning (CBR), where old cases to explain new situations or to justify new solutions are used. In [4], [5] a solution obtained through the cases relevant to the new problem is argued; while in [6] case explanation was able to explore particularities of the cases not explored by the rule-based inductive paradigm.

However, Artificial Neural networks (ANN) can not do the previous task, since in general they are not able to explain how a solution was reached. Hybrid systems are a

* This work was supported in part by VLIR (Vlaamse InterUniversitaire Raad, Flemish Interuniversity Council, Belgium) under the IUC Program VLIR-UCLV.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 238 – 248, 2006.

superior stage of KBS in which different technologies (ANN, CBR, genetic algorithms, et. al.) are combined to obtain a product that has their advantages and that minimizes their deficiencies [7], [8]. Some connectionist systems are presented in [7], in which symbolic modules allow the explanation. The "neurules" (an integration of symbolic rules with the Adaline neural unit) are used in [9] instead of symbolic rules, improving the efficiency of the inference mechanism while explanations in the form of "if-then" rules can be produced.

A model used to build hybrid KBS combining the connectionist and the CBR approaches is presented in [10] as a variant of the model Stanfill and Waltz [11]. The ANN is used for suggesting the value of the target attribute for a given query and the case-based module justifies the solution by retrieving similar cases. A neural net similar to Interactive Activation and Competition model is used [12]. The case-based module justifies the given solution using a similarity function. When a numeric attribute is used many different values for it in the case base should appear. Following the previous model, the quantity of neurons in the ANN very rapidly will be increased.

On the other hand, Neuro-Fuzzy computing constitutes one of the best-known visible hybridizations encompassed in soft computing, capturing the merits of fuzzy set theory and ANN. This integration promises to provide, to a great extent, more intelligent systems to handle real life ambiguous recognition [13], [14]. This paper presents an extension of the referred hybrid model [10] for the development of KBS, modeling numeric predictive attributes in terms of fuzzy sets [15]. Although fuzzy modeling usually comes two contradictory requirements in the obtained model: the interpretability and the accuracy, the model proposed follows the new tendency for an increasing of good balance between them [16].

The following section gives an overview of the former model referred. Section 3 describes in detail the proposed model using fuzzy sets, while experimental results are shown in section 4. Conclusions and future work in section 5 are presented.

2 Overview of the Original Model

ANN and CBR are two approaches of Artificial Intelligence that use the notion of similarity in an extensive way. According to [6], CBR is the essence of how human reasoning works. However, the massively parallel methods are preferred to CBR for their computational cost, but an ANN is a black box and usually we can not check whether its solution is plausible. The model presented in [10] (it will be referred as *original model*), combines these approaches to build a hybrid system. It takes advantage of the learning power of the ANNs and their capability to solve several types of problems, while CBR enables us to overcome their limited or null explanation capacity. The explanation based on cases consists in justifying the solution given to the problem by the ANN, presenting the cases similar to the problem and their solution.

Both approaches learn from examples. Let *CB* denotes a case base and $A=\{a_1, a_2,.., a_m\}$ the set of m attributes that describes an example e of *CB*. Let Da_i be the domain of the *ith* attribute and $i= 1\ldots m$. Each example e_k of *CB* is denoted by $e_k = (a_{1k}, a_{2k},.., a_{mk})$ and belongs a finite universe $Da_1 x\ Da_2$ x.. x Da_m (attribute space). If the index is immaterial, then an example will be written as $e= (a_1, a_2,.., a_m)$. A simple implementation of Interactive and Competition neural net, proposed

by Rumelhart in [12], is implemented. It is named SIAC. There is a group of neurons for each attribute. That is, each value in the domain Da_i of attribute a_i is represented by a single neuron. The relationships, which are joined by using a directed arc, are only considered between neurons of different groups. Each arc has associated a weight, whose value is achieved from examples in the *CB* by using a Hebbian-like learning mechanism [17]. The quantity of cases in which both values simultaneously appear is a measure of how these values are related.

The ANN completes the pattern corresponding with the query *q* taking into account the degree of activation of the neurons in the group of the target attribute *t*. This value is obtained by summing the weighted activations from the neuron corresponding with predictive attributes. The representative value T_j corresponding with the node of greatest activation will be the value given to this attribute. Later on, the CBR allows justifying the solution given by ANN with old experiences stored in the case base. The procedure to efficiently retrieve these cases uses a similarity function, in which the weights of ANN that store the necessary knowledge from the case base are used.

3 Extended Model Using Fuzzy Sets

In most of the cases, however, to represent a numeric attribute would be enough to take some values that represent a group of values close to them. In medical diagnosis, for instance, the values 15, 25, 35, 40, 50, 60, 75, 90 and 100 for "*age of the person*" could be taken as the set of representative values, according to the experts' criteria. Another way to select these values is via discretization [18], mapping the domain D_p of the attribute *p* into a somewhat arbitrary number of discrete ranges. Then, a representative value for one of intervals is selected. Both variants are centered on manipulation of numbers or symbols; and the traditional crisp set, in which an element is either present or not, should be followed by using the existing hybrid model. Nevertheless, when computational modeling uses words (e.g. *small*, *high*, *old),* natural language propositions are used as well as the brain's crucial ability to manipulated perceptions.

3.1 Numeric Attributes as Linguistic Variables

Measurements are crisp whereas perceptions are fuzzy [19]. Fuzzy set theory enables the use of natural language mapping numeric data into linguistic terms [20], [21]. Formally, the process by which individuals from a universe set are determined to be either members or non-members of a crisp set can be defined by a characteristic mapping [20]. This function can be generalized so that the values assigned to the elements of the universe set fall within a specified range and are referred to as the membership degree of these elements in the set. Larger values denote higher degrees of set membership. Such a function is called a membership function (MF).

When a numeric attribute $p \in P$ is modeled as linguistic variable, the values that appear for it in *CB* constitute the universe of the variable. Later, the set of linguistic terms $Rp = \{P_1, P_2, \ldots, P_{/Rp/}\}$ is considered the set of representative values of attribute *p,* where P_i represents the *i*th fuzzy set. As picking up the previous example for the attribute "*age of the person"*, the corresponding linguistic variable has real numbers as its universe and when turned into linguistic terms they may take values such as

'*young*', '*middle-age*' and '*old*'. For each of these terms a fuzzy set is defined through a MF, with all the advantages that this can represent [21]. Later on, the new structure of the case base provides a more natural framework to include expert knowledge by using linguistic terms.

Afterwards, how these membership degrees are used in the ANN model and in the similarity measure defining a new structure for the hybrid model will be explained in the followings topics 3.2 and 3.3 respectively.

3.2 The Neural Net with Fuzzy Sets

A neuron is allocated in the corresponding group for each linguistic terms defined. When a query q is presented, the processing neurons N_{Pi} corresponding with the ith representative value Pi for attribute p are either activated or not following the former model. However, if this attribute has been modeled using fuzzy sets then, a procedure to obtain the value p_{iq} as input to the neuron N_{Pi} is required. The way proposed in this paper is to modify the topology of the ANN of the *original model* by adding a preprocessor layer, where a preprocessor neuron is inserted for each representative value. It is named *Fuzzy-SIAC* (Fuzzy-Simple Implementation of the Activation and Competition model).

The preprocessing neuron is defined as a fuzzy neuron like type I referred in [22], which receives non-fuzzy inputs (the value of attribute p in the query q) and has associated a membership degree $\mu_{Pi}(p_q)$ as weight. The output of the preprocessing neuron corresponding with the ith representative value P_i is the result of the function f_{Pi}: $D_p \rightarrow [0, 1]$ that can be defined using (1) or (2). When the first expression is used, the proposed model is identified as the *Fuzzy-SIAC All* model. The Principle of Maximum Membership Degree [23] in expression (2) is applied (it is referred to as *Fuzzy-SIAC Max* model). Note that this variant of the new model proposed is the most similar one to the *original model* because only one neuron of the corresponding group in the ANN will be activated, but to a degree between 0 and 1.

$$f_{P_i}(p_q) = \mu_{P_i}(p_q) \tag{1}$$

$$f_{P_i}(p_q) = \begin{cases} \mu_{P_i}(p_q) & \text{if } \mu_{P_i}(p_q) = \max_{l=1}^{|R_p|} \mu_{P_l}(p_q) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where:

- p_q denotes the value given to predictive attribute p in the input pattern,
- R_p denotes the set of representative values of attribute p,
- $\mu_{Pi}(p_q)$ denotes the membership degree of the value p_q to the ith linguistic term P_i.

If the crisp set approach is followed to model an attribute p, now we turn to the particular case of the *original model* and the above can then be restated as follows:

- The set of the representative values $R_p \subseteq D_p$,
- The corresponding preprocessing neurons will be associated the weight $\mu_{Pi}(p_q)=1$ for all $P_i \in R_p$ and then the expressions define above can be used.

Considering again the attribute a as "*age of a person*" that takes values in the interval D_a= [15,110]. If the *original model* is followed, then the set of representative values R_a= {15, 40, 75} would be selected from to map its domain into the intervals A_1= [15; 35[, A_2= [35; 60[and A_3= [60; 110]. Otherwise, this attribute is modeled as a linguistic variable and R_a is defining as the set of linguistic terms mentioned above. This attribute will be represented in the ANN topology by using three processing neurons in both variants, but they will not receive the same activation when a query q is presented to the ANN. For example, if the age of patient takes value 40 for a query q (a_q = 40) and the *original model* is followed, then these neurons are activated with the values 0, 1 and 0 (see fig 1.A). Nevertheless in the last case, these values should be 0.2, 0.8, and 0 (see fig 1.B) according to the MF defined and considering all linguistic terms (see expression 1).

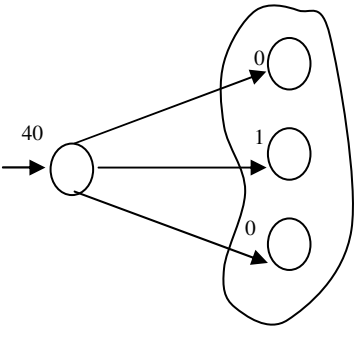


Fig. 1.A. Activation in the *original model*

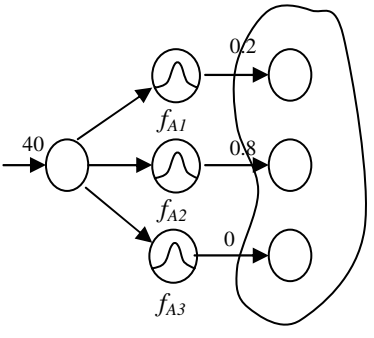


Fig. 1.B. Activation in the *Fuzzy-SIAC All* model

The same idea followed in the *original model* is applied to obtain the weights matrix. When non-supervised learning is applied, the modification of the weights takes place based on the states of the neurons (exits) after the presentation of certain stimulus (information of entrance to the network), without considering if it was desired to obtain or not these states of activation. Later, the new model should take into account the output of the preprocessing neuron.

Let a, b be attributes of A. The value labeled by $w_{Ai,Bj}$ represents the weight associated to the arc between processing neurons N_{Ai} and N_{Bj}. corresponding with their representative values A_i and B_j respectively. Specifically, a measure based on *Relative frequency* is used and then non-symmetric weight matrix is obtained. In other words, the value $w_{Ai,Bj}$ is obtained considering how the values taken by attributes a and b for each case e= (a, b) of CB are represented by their representative values A_i and B_j respectively, shared with how one of them is represented.

This expression according [24] is a systematic way of generating inclusion measures for ordinary sets in the form of a rational expression solely based on cardinalities of the sets involved. Its application to fuzzy approach based on the product t-norm to calculate the intersection of fuzzy sets and the cardinality as the simplest extension Σ-*count* [25] is proposed according the following expression.

$$w_{A_i,B_j} = \frac{\sum_{k=1}^{n} f_{A_i}(a_{ik}) f_{B_j}(b_{jk})}{\sum_{k=1}^{n} f_{A_i}(a_{ik})} \tag{3}$$

Note that the results depend on the expression chosen for the function *f*. For instance, if the Principle of Maximum Membership Degree is considered following expression (2), only the linguistic term that best represents the value of this attribute is considered. Finally, when a pattern corresponding with a query *q* is presented as input to neural net, the same procedure explained above to complete this pattern using the *original model* is applied. If the crisp approach is followed, then the *original model* is recovered.

3.3 The Similarity Function Using Fuzzy Sets

The CBR in the *original model* uses a similarity function to compute a measure of similarity between the pattern completed by the ANN and each example *e* in *CB*, referred to as objects *x* and *y* respectively, taking into account that:

a) Not all predictive attributes have the same importance
b) Two elements of the domain of an attribute can be equivalent for the problem of inferring the value of a target attribute
c) An Interpretative CBR is defined considering the value inferred for the target attribute.

Besides in the new model by using fuzzy sets to handle a predictive attribute *p*, a more accuracy measure than *original model* of "how close" is a value that *p* takes and P_i can be taking into account. In other words, in the new model the value that *p* takes in a query *q* will be represented for the representative value P_i in a value $p_{iq} = f_{Ai}(a_q)$ between 0 and 1 depending on $\mu_{Pi}(p_q)$. Afterwards, the value $f_{Ai}(a_q)$ and the knowledge stored in the ANN weights are taken into account to define:

Definition 1. The *strength* of the value a_x with respect to *T* is a measure of the influence of this predictive value in the activation of the target neuron representing the value *T* of the target attribute *t*.

$$S(a_x,T) = \frac{\sum_{\forall A_i \in R_a} w_{A_i,T} f_{A_i}(a_x) f_T(t_x)}{\sum_{\forall A_i \in R_a} f_{A_i}(a_x) f_T(t_x)} \tag{4}$$

where:

- a_x denotes the value of the attribute *a* in the object *x,*
- t_x denotes the value of the target attribute *t* in the object *x*
- A_i denotes the *i*th representative value associated with the attribute *a,*
- R_a denotes the set of representative values of the attribute *a,*
- $w_{Ai,T}$ denotes the weight of the arc between the processing neurons N_{Ai} and N_T,
- *f* is the function defined in the previous section.

Definition 2. The (degree of) *equivalence* between two values a_x and a_y for predictive attribute a in the context of target attribute t is defined as:

$$\delta_t(a_x, a_y) = 1 - \frac{\sum_{\forall T \in R_t} |S(a_x, T) - S(a_y, T)|}{|R_t|} \tag{5}$$

where: R_t denotes the set of representative values (linguistic labels) of the target attribute t.

Definition 3. The *importance* of predictive attribute a for object x in the context of target attribute t is defined as its relative strength among all predictive attributes:

$$I_t(x, a_x) = \frac{S(a_x, t_x)}{\sum_{\forall p \in P} S(p_x, t_x)} \tag{6}$$

Definition 4. The *similarity* between objects x and y, is the weighted sum of equivalences, using importance as weights:

$$\beta(x, y) = \sum_{p \in P} I_t(x, p_x) \delta_t(p_x, p_y) \tag{7}$$

where x and y belong to the attribute space defined before. Note that expressions (5) and (6) allow defining a general expression independent of particular domain because both expressions use the concept of strength defined in expression (4). The ANN weights are computed from examples about the specific domain, which are used to define the equivalence and importance for all types of attributes regardless of the differences between their domains. Although above expressions are referred to a single target attribute to make its easily understanding, they can be easily generalized for more than one target attribute.

The most similar cases recovered allow the user to understand the solution given by the ANN. This is in fact the "justification". In other words, it is possible the interpretation of the knowledge encoded in numbers as ANN weights by means of attributes values contained in such cases. For that reason, terms defined by expressions (6) and (7) are relative to a target attribute t, and they guarantee take into account only information relevant to a specific problem.

4 Experimental Results

In the Experiments, nine international datasets from UCIMLR [26] are used. They only have numerical attributes, without missing values and one target attribute. A discretization method Equal Width and trapezoidal membership functions developed from these same intervals were used to select a set of representative values for each attribute.

A simple measure of the performance for the ANN and CBR by using expressions (8) and (9) respectively can be achieved. In the first expression we consider the

expected value t_q for the target attribute t in the query q of the control set *CS*, while t'_q refers to the value inferred by the ANN in both expressions. In the second expression t_e refer to the value of the target in the most similar case e *(*1-nearest neighbor or 1-NN*)* selected after computing the similarity function $\beta(q,e)$ for all e of *CB*. Finally function 1( ) yields 1 iff its argument is true.

$$Performance ANN(CS) = \frac{\sum_{\forall q \in CS} 1(t_q = t'_q)}{|CS|} \tag{8}$$

$$Performance CBR(CS) = \frac{\sum_{\substack{\forall q \in CS \\ \forall e \in kq}} 1(t'_q = t_e)}{|CS|} \tag{9}$$

Three variants previously defined in order to compare the performance of the new model with the former one (*original model*) are considered. A comparison of the paired samples was carried out, using both non-parametric tests: the Friedman two-way Analysis of Variance and the Wilcoxon signed-rank test.

A 10-fold cross validation was applied, and the performance mean (m) and the performance variance (s^2) with the ANN and CBR module considering all cases in the control data for each data set were calculated. The following Table 1 shows these values referred to the ANN of the *original model*, and the new one in both *Fuzzy-Max* and *Fuzzy-All* variants considered.

Table 1. The performance of the ANN using variants previously defined

Dataset names	*SIAC Original Model*		*Fuzzy-SIAC Max*		*Fuzzy-SIAC All*	
	m	s^2	m	s^2	m	s^2
Iris	89.33%	0.06	89.33%	0.06	94.00%	0.05
Diabetes	65.10%	0.05	65.10%	0.05	65.10%	0.05
Glass	43.41%	0.05	46.78%	0.09	39.26%	0.06
Liver-disorders	58.02%	0.13	58.02%	0.13	58.02%	0.13
Vehicle	50.83%	0.07	50.69%	0.07	51.29%	0.07
Wine	89.54%	0.09	96.53%	0.05	80.88%	0.12
WBC	89.90%	0.03	89.90%	0.03	88.28%	0.03
Ionosphere	64.13%	0.08	64.13%	0.08	64.13%	0.08
Segmentation	73.86%	0.03	74.24%	0.04	76.97%	0.06

The analysis of the results shows that the performance mean and variance of the ANN in the three variants considered do not have a significant difference (significance of the test: 0.957 and 0.845 respectively). Then, considering previous explanations, the three hybrid models compared don't have significant differences of performance.

The performance achieved with the interpretative CBR using expression (9) always is one using the similarity function defined above. In other words, the target value of

the most similar case recovered to a query q is the same value given by the ANN. For instance, consider an example of "Iris" dataset that belongs to class *´Iris virginica´* and it is correctly classified by the ANN. Table 2 shows the most similar cases recovered (only its index in the case base) and the similarity measure achieved, which is considered as "justification" given by these tested hybrid models. In other words, the information provided by these cases allows explaining the inference given by the ANN, instead of numeric knowledge stored in the ANN that really was used to solve the problem. Note that all cases recovered are in the same class inferred for the problem, but the model proposed give a gradual similarity measure considering the difference between the cases recovered with the problem solved, which is not possible using the *original model*.

Table 2. The most similar cases for an example of the Iris data

Original Model		*Fuzzy-Max*		*Fuzzy-All*	
Index	*value*	*Index*	*value*	*Index*	*value*
103	1.00	118	0.977	106	0.989
106	1.00	106	0.976	118	0.985
110	1.00	132	0.971	132	0.982
118	1.00	119	0.959	119	0.978
119	1.00	136	0.943	131	0.970
125	1.00	108	0.925	136	0.968
132	1.00	131	0.922	108	0.967
136	1.00	126	0.905	126	0.956

5 Conclusions and Future Work

This paper has presented an extension of an existing hybrid model for the development of knowledge-based systems, modeling of predictive attributes in terms of fuzzy sets. The ANN is used for suggesting the value of the target attribute for a query given. Afterwards, the case-based module justifies the solution provided by the ANN using a similarity function, which includes the weights of the ANN and the membership degrees in the fuzzy sets considered.

A system developed by using the proposed model and the former one will not have significant differences of performance, preserving the advantages of the hybridization used in the *original model*, guaranteeing robustness and interpretability; while a more natural language than in former one is used. In other words, the proposed model allow to avoid the null explanation of the connectionist approach, providing a natural framework to include expert knowledge by using linguistic terms when numeric attributes are used.

The performance of the hybrid model proposed depends on the parameters of the MF used. As future work, the solution given by the ANN to cases of the case base and some criteria from human experts can be considered to define an algorithm to tune these parameters. It could allow to increase the accuracy of the system developed using the proposed model.

References

1. Ganascia, J.G.: Explanation and Justification in Expert System. Computers and AI, Vol. 4 #1 (1995): 3-13
2. Montani, S., Magni, P., Roudsari, A.V.,Carson, Bellazzi, R.: Integrating Different Methodologies for Insulin Therapy Support in Type 1 Diabetic Patients. Lecture Notes in Computer Science, Vol. 2101. Springer-Verlag, Cascais, Portugal (2001): 20-27
3. Kolodner J.: An introduction to case-based reasoning. Artificial Intelligence Review 6 (1992): 3-34
4. Kass, A.M. et. al: Case-based reasoning applied to constructing explanations, in: Proceeding of workshop on case-based reasoning . Florida, 1988
5. Kass, A.M; et. al: SWALE: A program that explains, in Explanation Patterns Understanding Mechanically and Creatively.1986
6. Lopes, A.A., Alipio, J.: Integrating Rules and Cases in Learning via Case Explanation and Paradigm Shift. Springer-Verlag (2000)
7. Fdez-Riverola, F., Corchado, J.M.: Sistemas híbridos Neuro-Simbólicos: Una revisión. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No 11, Vol I (2000): 12-26
8. Prentzas, J., Hatzilygeroudis, I.: Integrating Hybrid Rule-Based with Case-Based Reasoning. Lecture Notes in Artificial Intelligence. SpringerVerlag (2002): 336-349
9. Hatzilygeroudis, I. et. al.: Neurules: Improving the performance of symbolic rules. International Journal on Artificial Intelligence Tools (IJAIT), vol. 9, No. 1. 2000
10. García, M.M., Bello, P.R.: A model and its different applications to case-based reasoning. Knowledge-based systems 9 (1996): 465-473
11. Stanfill, C., Waltz, D.: Toward memory-based reasoning. Comm. of ACM, 29 (1986): 1213-1228
12. McClelland, David E. Rumelhart: Explorations in parallel distributed processing. MIT Press (1989)
13. J.-S. R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall (1997)
14. Pal, S.K., Dillon, T.S., Yeung, D.S. : Soft Computing in Case Based Reasoning. Springer Verlag London (2001)
15. Zadeh, L.A.: The concept of a linguistic variable and Its Application to Approximate Reasoning. Information Sciences Vol. 8 (1975): 199-249
16. Casillas, O. Cordón, F. Herrera, L. Magdalena: Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview. Interpretability issues in fuzzy modeling. Vol. 128. Springer (2003)
17. Sima, J. : Neural Expert System. Neural Networks. Vol 8 #2 (1995)
18. Kurgan L.; et. al. CAIM Discretization Algorithm. IEEE Transactions on Knowledge and Data Engineering. Vol 16. No. 2. 2004.
19. Lotfi A. Zadeh: From Computing with Numbers to Computing with Words -From Manipulation of Measurements to Manipulation of Perceptions. Intelligent Systems and Soft Computing (2000): 3-40
20. Hong, J., Wang, X.: On the handling of fuzziness for continuous-valued attributes in decision tree generation. Fuzzy Sets and Systems Vol. 99, Issue 3 (1998): 283-290
21. García, MM., Rodriguez, Y., Bello, R.: Usando conjuntos borrosos para implementar un modelo para sistemas basados en casos interpretativos. In Proceedings of IBERAMIA-SBIA. Eds por M. C. Monard y J.S. Sichman, Sao Paulo, Brasil, (2000)

22. Len, C.-T., Lee, C.S.: Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent System. Prentice Hall (1999)
23. Nauck, D., Klawonn, F., Kruse, R.: Foundations of Neuro-Fuzzy Systems. John Wiley & Sons Ltd (1997)
24. De Baets B., De Meyer, H., Naessens, H.: On rational cardinality-based inclusion measures. Fuzzy Sets and Systems 128 (2002) : 169-183
25. Zadeh, L.A.: A Computational Approach to Fuzzy Quantifiers in Natural Language. Computing and Mathematics with Applications, 9(1) (1983): 149–184
26. Murphy, P.M., Aha, D.W.: UCI Repository of Machine-Learning Databases, http:\\www.ics.uci.edu/~mlearn\mlrepository.htm

Development of a Neural Sensor for On-Line Prediction of Coagulant Dosage in a Potable Water Treatment Plant in the Way of Its Diagnosis

Hector Hernandez[1] and Marie-Véronique Le Lann[1,2]

[1] Laboratoire d'Analyse et d'Architecture des Systèmes, LAAS-CNRS
7, Avenue Colonel Roche, 31077 Toulouse cedex 4, France
{hhernand, mvlelann}@laas.fr
http://www.laas.fr
[2] Université de Toulouse, INSA, Département de Génie Electrique et Informatique
135, Avenue de Rangueil 31077 Toulouse cedex 4 France
http://www.insa-toulouse.fr

Abstract. Coagulation is one of the most important stages in surface water treatment, allowing for the removal of colloidal particles. Its control, in the majority of the plants remains still manual and requires long and expensive analyses of laboratory which provide only periodic information. The present work describes an innovative methodology integrating various techniques for the development of a diagnosis system of this plant. A first part, concerned by this paper, consisted in developing a software sensor based on artificial neural networks for predicting on-line the amount of optimal coagulant dosage from raw water characteristics such as turbidity, pH, temperature, etc. In a second part this information will be integrated like an input to a diagnosis system of the plant. The development of the neural sensor has been performed based in real data covering several years of operation and its performance have been compared toward those given by a multi-linear interpolation.

1 Introduction

Water industry is facing increased pressure to produce higher quality treated water at a lower cost. The coagulation-flocculation is a major step in the production of potable water, allowing the removal of colloidal particles. The coagulant dosing is not only one of the major control parameters in the coagulation process, but represents also the major operation cost in a water treatment plant. Most coagulant dosing is determined by the way of jar test (off-line dosing tests). However, the jar test can only provided periodic operation information, which can not be applied to real-time control of the coagulation process, especially with a time-varying raw water quality (principally during a spring-summer runoff period). An excessive coagulant overdosing leads to increased treatment costs, while an underdosing leads to a failure to meet the water quality targets and a less efficient operation of the water treatment plant. During the last decade a number of models based on artificial neuronal networks (ANN) have been developed and applied for predicting the coagulant dose concentrations in a

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 249 – 257, 2006.

water treatment process (Baba, 1990 [1]; Mirepassi et al., 1995 [2]. A fuzzy neural system, coupling of ANN with fuzzy theory, has been applied to extract the control rules by learning from historical operational data of water treatment process [3].

This paper addresses the problem of automatic coagulation control based on the raw water characteristics such as turbidity, pH, temperature, etc. Some recent studies (Valentin [4]; Lamrini et al., 2005 [5]) show the potential effectiveness of such an approach based on ANN's by means of the implementation of a neural software sensor for on-line prediction of the coagulant dosage.

The innovative aspect of this work resides in the design of a neural sensor integrating various techniques to firstly determine the set of accurate measurements needed to predict the coagulant dosing rate and to reduce this set to the main variables with a view to integrate the coagulant dosing rate which will be on-line computed by the neural sensor as an input of a diagnosis system, which should allow the portability of the system at a low cost from one site to another.

A brief description of the water treatment plant and the operation units involved in the drinking water treatment process is first provided in section 2. The methodology used for the design and synthesis of the Neural Software Sensor is given in section 3. Finally, experimental results and a comparison with a linear interpolation model are presented and discussed in section 4.

2 Water Treatment Process

2.1 Overview of Water Treatment Operations

The water is the most abundant compound on the surface of the world. Water treatment involves physical, chemical and biological processes that transform raw water into drinking water which satisfies a whole of quality standards at a reasonable price for the end-user.

The "SMAPA" water treatment plant (Tuxtla city, Mexico), which was used as an application site for this study, provides water to more than 800,000 inhabitants and has a nominal capacity to process 800 l/s of water per day. The figure 1 presents a schematic overview of the various operations necessary to treat the water, the available measurements, and the coagulant dosing point. The complete usual chain comprises the 5 great following units: pre-treatment, pre-oxidation, clarification, disinfection, and refining. The present work concerns essentially the coagulation process which represents the main step in the treatment process.

Raw water is collected at the river "Grijalva" and pumped to the treatment plants. Water treatment plants invariably include two main process units, clarification and filtration. Other units may be required depending of the quality of the water source.

The coagulation process is brought about by adding a highly ionic salt (aluminum sulfate) to the water. A bulky precipitate is formed which electrochemically attracts solids and colloidal particles. The solid precipitate is removed by allowing it to settle to the bottom of the tank and then periodically removing it as sludge. The coagulation process accounts for the removal of most of the undesirable substances from the raw water and hence a tight monitoring and control of this process is essential. The next stage is filtration, where the particles passing trough the previous stages are removed.

The final stages in the process are chlorination and pH adjustment. The water is then stored in a tank and ready to be transported through the water supply network.

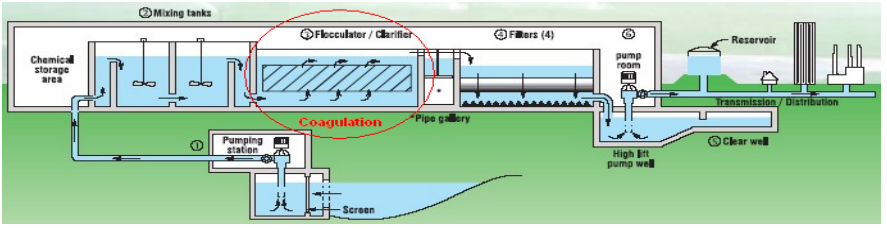


Fig. 1. The drinking water plant

2.2 Coagulation Control

Surface waters contain both dissolved and suspended particles. The suspended particles vary considerably in source, composition charge, particle size, shape and density. The correct design of a coagulation process and the selection of coagulants depend on understanding the interactions between these factors. This process is one of the most important stages in a surface water treatment, allowing for the removal of colloidal particles. The main difficulty is to determine the optimum quantity of chemical reagent based on the raw water characteristics. The traditional method of controlling coagulant dose, called the jar-test, relies heavily upon human intervention and leads to poor control with wastage of expensive chemicals or failure to meet the water quality targets. It involves taking raw water samples and applying different quantities of coagulant to each sample [6, 7]. Disadvantages associated with such a procedure are the necessity to rely on manual intervention, and the lack of adaptation to abrupt changes of water characteristics. The objective of this paper is to provide a complementary support to the jar-test allowing for the automatic determination of optimal coagulant dose from raw water characteristics, using an artificial network approach. This approach requires the availability of on-line water quality measurements.

3 Methodology of the Neural Software Sensor Design for Its Inclusion in the System Diagnosis

The coagulation process is difficult to model using traditional models. The coagulant dosage ensuring optimal treatment efficiency has been shown experimentally to be non-linearly correlated to raw water characteristics which are usually available on-line. The development of the prediction system of the optimal coagulant dosage was performed following two steps: a pre-treatment step using a Principal Components Analysis to determine the set of needed measurements and then the design of Artificial Neural Networks (ANN) for the prediction of the coagulant dose using as inputs the set of measurements identified in the former step.

3.1 Principal Components Analysis (PCA)

The design of an ANN includes several steps: the choice of the architecture and the determination of its parameter. The former step involves the determination of the ANN inputs. These inputs must hold enough information for a modeling accuracy but must not be redundant or useless to prevent from an over parameterized model. PCA is one of the multivariate methods of analysis which has been used widely with large multidimensional data sets. More details on PCA can be found in [8]. The full set of principal components is as large as the original set of variables. But it is common to restrict the number of principal components to the first few principal components which exceeds 80% of the total variance of the original data. This allows replacing the set of original variables by these first few variables.

3.2 Artificial Neural Networks (ANN's) and Multi-layer Perceptron (MLP)

ANN's are one of the earliest adaptive techniques in engineering and computing science. An ANN is a network of neurons or processing elements and weighted connections [9]. ANN is fundamentally a mathematical model composed by a set of nodes (artificial neurons) where information is processed. Once its architecture has been chosen, a second step is the determination of its parameters (the weights attached to the connections between the nodes). This step is purely an identification of nonlinear model parameters denoted as the learning phase. Since we dispose of historical data (input-output patterns which compose the learning base), the supervised learning has been used for the determination of the weights. For each input pattern, the network generates an output pattern, which is compared with the desired output (also called target) and the adaptation of the model parameters is made in relation to the observed error [9,10]. One of the most studied and used ANN architecture is the MLP.

The prediction of optimal coagulant dosage from water characteristics is a non linear regression problem which can be tackled using MLP's. It consists in an input-output network of which the neurons are distributed according to several layers, fully connected between adjacent layers, and where the flow of information is done in a feed-forward way. The MLP is usually trained by gradient descent methods [10], in which the error is propagated backwards through the network. Figure 2 shows a MLP

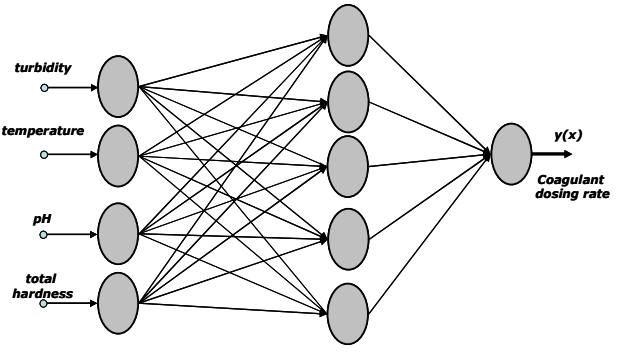


Fig. 2. The Multi-Layer Perceptron (MLP)

with three layers: an input layer (characteristic variables of the raw water quality) with n neurons, a hidden layer with H neurons and a layer with one output neuron (the computed optimal coagulant dosing rate).

The expression of the output of MLP is given by:

$$y(x) = f\left(\sum_{j=1}^{H} W_j h_j + W_0\right) \quad and \quad h_j = f\left(\sum_{i=1}^{n} w_{ji} x_i + w_{j0}\right) \tag{1}$$

where w_{ji} are the weights between the input layer and the hidden layer and W_j the weights between the hidden layers and the output layer, f is the activation function, supposed to be the same for all the neurons. The transfer function f can be any function [11], but for most practical uses of neural networks it is important to have a continuous, completely differentiable function. In our case, we know that the relationship between the coagulant dosage with the raw water characteristics is non-linear so the choice of a sigmoid transfer function has been made.

The method traditionally used to perform the training of such networks, e.g. to adjust the weighted connections, is the Backpropagation learning algorithm [10]. There are a number of variations on the basic algorithm which are based on other standard optimization techniques, such as conjugate gradient, Newton and Levenberg-Marquardt methods. Learning occurs when the network emulates the non-linear function underlying the training data set.

As previously mentioned, the design of ANN has been performed through 2 steps: the determination of the architecture and then the identification of its parameters (the weights). Concerning the first step, the number of neurons in the input layer has been determined by the PCA application (see section 4.2). So only 4 input neurons are used: Turbidity, Total Hardness, Temperature and pH. The number of hidden neurons has been determined by an iterative procedure regarding the performance both in training and test. Firstly, for a given number of neurons in the hidden layer, training is stopped when the MSE value does not increase anymore (figure 3). Then the number of neurons in the hidden layer is increased until the MSE becomes minimal (figure 3). As it can be noticed in figure 3, 50 neurons in the hidden layer are sufficient and give a good compromise between complexity and generalisation performance.

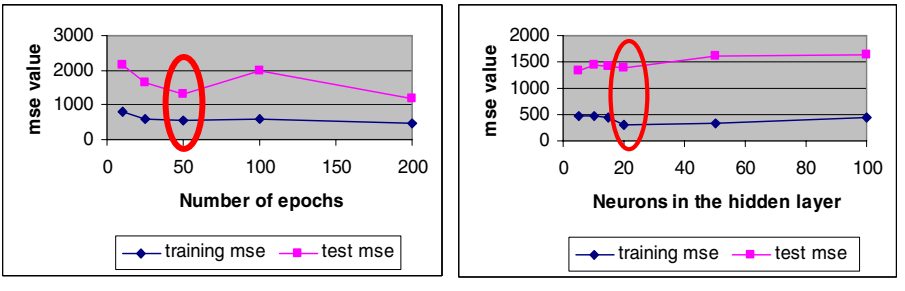


Fig. 3. MSE as a function of epochs and number of neurons in the hidden layer

4 Application of the Method to the SMAPA Plant

The raw database consists of 364 measurements of 9 variables during a period of 12 months (January to December 2002). Every sample underwent to different physical and chemical analysis as well as to the jar-testing in order to determine the coagulant dosage.

4.1 Selection of Describers

For this study, 9 describers of the raw water quality are adopted: temperature (T), color (C), turbidity(TUR), total solid suspends (TSS), organic mater (OM), pH, bicarbonate (B), chloride (CL), total hardness (TH), and coagulant dose (DOSE) (as a passive parameter). The figure 4 presents the evolution of 4 different describers of the raw water quality with time.

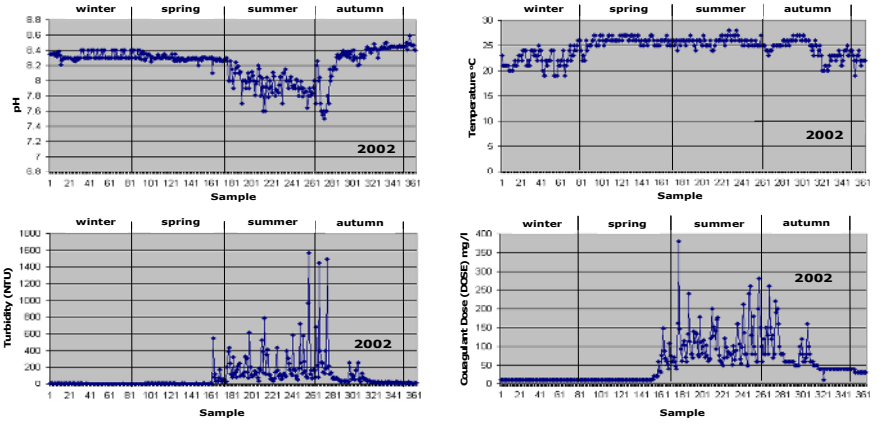


Fig. 4. Raw water characteristics of the SMAPA plant (4 describers)

A PCA applied to these data yield the following results presented in figure 5 where a fast steep decrease of the eigen values can be noticed. Only the first four components appear useful to the analysis while representing more than 88 % of the inertia.

	Eigen value	%total variance	cumul %total variance
1	5.446	54.455	54.455
2	1.812	18.116	72.571
3	1.010	10.099	82.670
4	0.535	5.354	88.024
5	0.420	4.197	92.221
6	0.250	2.500	94.721
7	0.218	2.180	96.901
8	0.154	1.541	98.442
9	0.090	0.900	99.342
10	0.066	0.658	100.000

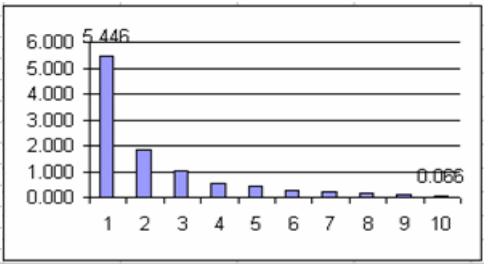


Fig. 5. The results of the PCA

The first component is predominant because it explains 54.45% of the information contained in the 10 initial variables.

The second, third and fourth component have a medium and equivalent importance: they represent respectively 18.11, 10.09 and 5.35% of the total inertia. Therefore, we can get a good representation by keeping only the four first principal components. The set of the 10 variables can then be simplified and replaced by 4 new variables represented by these components. Table 1 shows the contribution of each principal component to the total inertia and the relationship variable-component.

Table 1. Contribution to the total inertia and relationship variable-component

Variable	Comp.1	Comp.2	Comp.3	Comp.4
Temperature (T)	0,131	0,229	0,863	0,180
Color (C)	0,354	0,250	-0,232	0,236
Turbidity (TUR)	0,349	0,251	-0,270	0,306
Total Solid Suspends (TSS)	0,393	0,102	0,097	-0,116
Organic Mater (OM)	0,374	0,109	-0,159	0,062
pH	-0,216	-0,176	-0,207	0,616
Bicarbonate (B)	0,214	0,506	-0,191	-0.479
Chloride (CL)	-0,267	0,455	0,084	0.436
Total Hardness (TH)	-0,274	0,540	-0,066	-0,034
Coagulant dose (DOSE)	0,375	0.121	-0.057	-0.006

The interpretation of the results is as follows: the first component that represents 54.45% of the total inertia is positively defined by 5 variables much grouped (C, TUR, TSS, OM and DOSE). The second component (18.11%) is defined with 3 variables much grouped (B, CL and TH). The third component (10.09%) represents the temperature variable whereas the fourth (4.35%) is correlated with the pH variable. Our study consists in the prediction of the coagulant dosing rate from the raw water characteristics according to the descriptors easily measurable in continuous. According to the previous results, it is clear that only one of the four variables constituted the first component group (obviously without the dose) can be chosen as a measure variable. A similar reduction can be done concerning the variables constituting the second component. The choice of these two variables has been made according the facility of on-line measurement on the plant: in our case we keep only the following variables: TUR, TH, and the two others variables T and pH directly correlated to the third and fourth component respectively.

4.2 Prediction of Coagulant Dosage

Thus, the number of input neurons is 4 (TUR, TH, TEMP and pH). The training was carried out over the first year (2002). The average error represented by the Matlab criterion *MSE* [12] calculated by the network is 0.085 on the training set corresponding to data of year 2002. The validation of the ANN has been performed on test data of year 2003 not included in the initial training set: the criterion *MSE* is a little higher

(0.092). Figure 6 shows the prediction accuracy of the ANN model for this validation set of year 2003. The predictions given by neural network (point line) are very close to the real data.

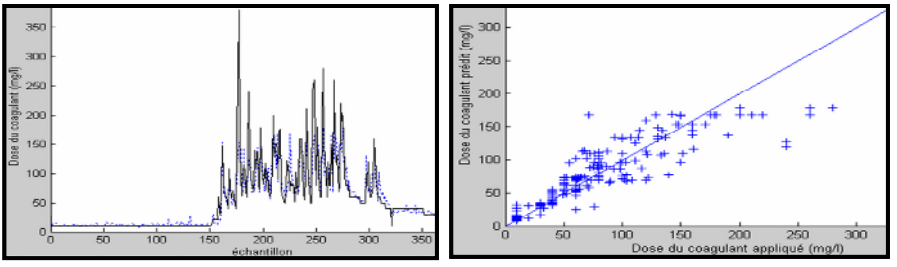


Fig. 6. Left: actual (thin line) versus predicted (point line) coagulant dosage with ANN model on test data. Right: predicted versus target coagulant dose

4.3 Linear Regression

The PCA enables also to get a multi-linear model. It is thus possible to derive a multi-linear decomposition of the dose versus the same input variables as for the ANN. Figure 7 shows the outputs of the linear model trained with the same data as the ANN. As expected, the prediction accuracy is clearly poorer than the one of the neuronal model Performance results obtained from this data set with ANN model and linear regression are presented in table 2.

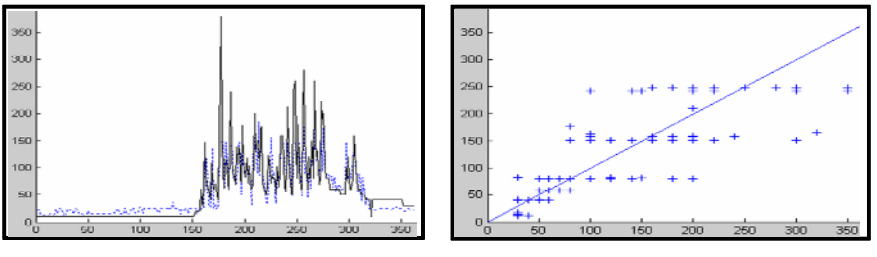


Fig. 7. Left: actual (point line) versus predicted (thin line) coagulant dosage with linear regression on test data. Right: predicted versus target coagulant dosage

Table 2. Compared results for ANN and linear regression models

Comparison indexes	**ANN model**	**Linear regression model**
R^2 for data training	0.97	0.72
R^2 for data test	0.96	0.61
Performance MSE for data training	619.7	859.3

5 Conclusions

A neural software sensor for coagulation control was developed from Artificial Neural Networks. It supplies us on real time the coagulant dosage to inject in the coagulation unit which is the key stage of the process in a water treatment plant. The selection of network inputs was made by using the statistical technique PCA to allow eliminating the redundant information. Experimental results using historical real data have demonstrated the efficiency of this approach. A multilinear regression model was also developed from the PCA for comparison with the neural network model.

Water treatment units contain complex processes. Few researches so far concerned their control or diagnosis. However, in front of the more and more necessity of producing a water of constant quality, the producers of dinking water becomes more sensitive to any technique allowing answering quickly this requirement. For that purpose, the future works aim at establishing a methodology of diagnosis based on the interpretation of the information obtained of the entire water treatment plant by including as important information the value of the coagulant dosage calculated by the neural software sensor.

References

[1] K. Baba, I. Enbutu, M. Yoda. *Explicit representation of knowledge acquired from plant historical data using neural network.* Int Joint Conf on Neural Networks. Wash, D.C., 1990.

[2] A. Mirsepassi, B. Cathers, H.B. Dharmappa. *Predicted of chemical dosage in water treatment plants using ANN and Box-Jenkings models.* Preprints of 6th IAWQ Asia-Pacific Regional Conference. Korea, 1997, 16-561-1568.

[3] I. Enbutsu, K. Baba, N. Hara, K. Waseda, A. Nogita. *Integration of multi AI paradigms for intelligent operation support system – fuzzy rule extraction from a neural network.* Wat. Res. 1998, 28, 11-12.

[4] Valentin, N.: *Construction d'un capteur logiciel pour le contrôle automatique du procédé de coagulation en traitement d'eau potable*. Thèse de doctorat, UTC/L.desEaux/CNRS (2000)

[5] B. Lamrini, A. Benhammou, M-V Le Lann , A. Karama *A neural software sensor for online prediction of coagulant dosage in a drinking water treatment plant*, Transactions of the Institute of Measurements and Control, **27**, N°3 pp. 195-213 (2005)

[6] C. Lind. *Coagulation Control and Optimization: Part One*, Public Works, Oct 1994, 56-57

[7] C. Lind. *Coagulation Control and Optimization: Part Two*. Public Works, Nov 1994, 32-33

[8] J.E. Jackson. *A user's guide to principal components*. Wiley Series in Probability and Mathematical Statistics, John Wiley &Sons, (2003)

[9] W.S. McCulloch, W. Pitts. *A logical calculs of the ideas immanent in nervous activity*. Bulletin of Math. 1943, Biophysics, 5, 115-133.

[10] D.E. Rumelhart, J.L. McClelland. *Parallel distribution processing: exploration in the microstructure of cognition*. Cambridge, MA, 1986, MIT Press, 1.

[11] W. Duch. *Survey of neural transfer functions*. Neural Computing Surveys, 1999, 2.

[12] Matlab, *Neural Network Toolbox*, User' Guide, Inc., (2001)

Multi-objective Memetic Algorithm Applied to the Automated Synthesis of Analog Circuits

Maria José P. Dantas[1,3], Leonardo da C. Brito[2], and Paulo Henrique P. de Carvalho[3]

[1] Department of Mathematics and Physics, Catholic University of Goiás
Av. Universitária 1069, Setor Universitário, CEP 74605-010, Goiânia, Goiás, Brazil
mjpd@cultura.com.br
[2] School of Electrical and Computer Engineering, Federal University of Goiás
Av. Universitária s/n, Setor Universitário, CEP 74695-220, Goiânia, Goiás, Brazil
brito@eee.ufg.br
[3] Department of Electrical Engineering, University of Brasília
Campus Universitário Darcy Ribeiro, Asa Norte, CEP 70910-900, Brasília–DF, Brazil
paulo@ene.unb.br

Abstract. In this paper, a Multi-objective Memetic Algorithm applied to the automated synthesis of analog circuits is proposed. The optimization of circuit topologies and their parameters are simultaneously carried out. A variable-size 2D circuit representation is used. In this approach, the initial solutions are created based on expert knowledge through the use of well-known building-blocks and rules-based coupling schemes. The proposed genetic operators are specific to 2D encoding and they are capable of fomenting a balance between diversity and convergence. A local search process – the Simulated Annealing method – is applied in order to improve the circuit parameters. The results show that the proposed method generates, with small populations and few generations, small well-structured circuits which accomplish the specifications.

Keywords: multi-objective memetic algorithm, automated synthesis, 2D representation, building-blocks, analog circuit.

1 Introduction

Evolutionary circuit synthesis is a very difficult task. This process presents peculiar features since it concerns the optimization of both circuit structure (topology) and its parameters (values to be assigned to the circuit components, as a resistance, for example). Thus, the objective is to search for good circuit topologies and their associated parameters, in order to achieve the desired goals. It is required an explicit search control for structures and for parameters. The problem is how to balance the exploration/exploitation on the topology and parameters search. We have to avoid promising structures to be discarded because their parameters are not tuned well enough to show their potential. On the other hand, we also have that avoid that weak structures with better parameters proliferate and dominate the entire population, thus leading to premature convergence.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 258 – 267, 2006.

Using evolution-based paradigms (Genetic Algorithms and Genetic Programming), researchers (see [1]-[4] for example) have been able to evolve both circuit topologies and components values, without providing any prior specific design input to the algorithm. That is, they do not require expert knowledge about the circuit topology. However, these methodologies have some basic drawbacks. A huge amount of completely invalid (anomalous) circuits are generated along the synthesis process, thus increasing the time required for achieving good solutions. On the other hand, they commonly generate extremely unconventional and unstructured circuit topologies, which can be physically unpractical. Besides that, they demand high computational resources and suffer from premature convergence [1].

In this paper, we present a hybrid evolutionary method to overcome these problems. The main elements of the proposed methodology are: (1) expert knowledge to place a set of moderate constraints on the structure of the candidate solutions, in order to reduce the search space and to avoid anomalous circuits, but with enough flexibility to allow the generation of novel topologies; and (2) a Memetic Algorithm [5] to balance the topology search (performed by a Evolutionary Algorithm) and the parameters tuning process (performed by the Simulated Annealing method). The hybrid algorithm works with suitable circuit representation and genetic operators.

This paper is divided into four sections. In section 2, the proposed method is described. In section 3, we present a synthesis experiment and discuss the results comparing with other results reported in the literature. Section 4 offers concluding remarks and presents the perspectives for future works.

2 Description of the Proposed Method

We describe in this section the Multi-objective Memetic Algorithm with variable-size 2D representation. It is appropriate for evolving/refining solutions, using small populations and few generations, and for producing small well-structured circuits. The flowchart of the hybrid Evolutionary Algorithm is shown in Fig. 1. The details about the main elements of the hybrid algorithm are given as following.

Circuit Representation. An unconstrained circuit representation generally produces a large number of anomalous circuits along an evolution process, which increases the time required for convergence [1],[6]. On the other hand, a constraining representtation reduces the probability of finding unconventional circuit structures, but it reduces the search space. Then, a better representation must impose a moderate set of constraints on the structure of a generated circuit. In this work, the expert knowledge obtained in traditional synthesis [7]–[10] is used as intelligent input. The use of building-blocks associated with rules-based coupling schemes provides a means for accelerating the evolutionary process, achieving regular well-structured solutions. The building-blocks used are small lumped-elements circuits (combinations of capacitors and inductors). They are available in a database and treated as black-boxes by the algorithm. The internal topology of a building-block cannot be changed. The rules-based coupling schemes allow us to connect well-known building blocks while

keeping the circuit structured and then avoiding the occurrence of anomalous topologies.

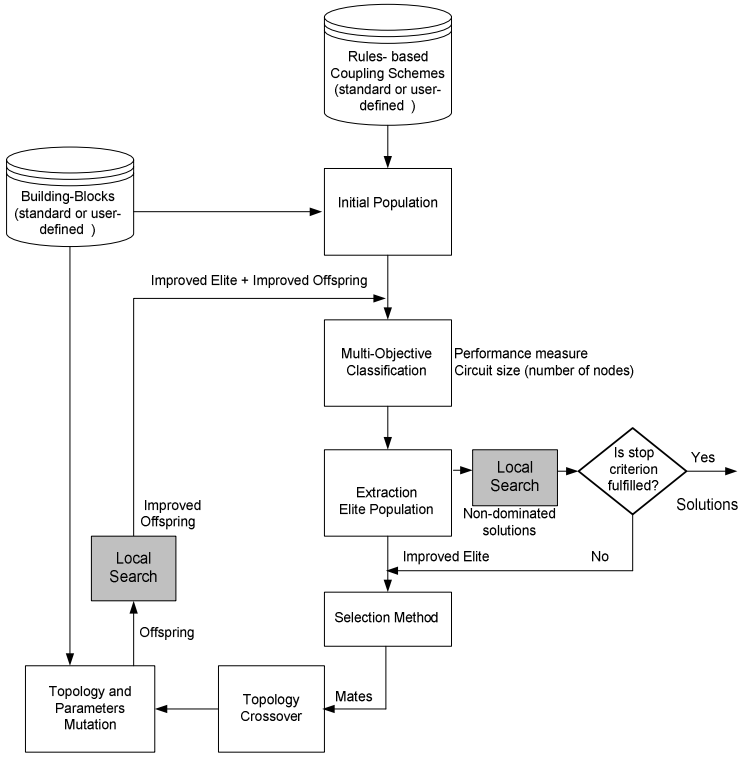


Fig. 1. Flowchart of the Multi-objective Memetic Algorithm specialized for the optimization of both structure and parameters of analog circuits

The proposed method uses the template circuit shown in Fig. 2. The circuit representation corresponds to an n-node undirected graph. So, it is enough an upper triangular matrix (hereinafter referred to as reduced matrix) for representation. The reduced matrix is directly handled by the algorithm without being necessary an equivalent linear encoding scheme (1D). In [11], a 2D matrix representation for circuit synthesis was proposed. The authors also proposed a 2D-to-1D transformation applied before the crossover process. However, as evidenced in [12], 2D-to-1D mapping results in loss of neighboring information between elements of the structure (a circuit, in this case) which can be harmful for the evolution process. The reduced matrix has the same size as the number of circuit nodes. Each entry (i, j) of the matrix represents a pair of external nodes which connect the two terminals of a building-block. Notice that $i = j$ means that one terminal of the building-block is connected to the node i, and the other terminal is connected to the reference ground (node 0). This matrix does not allow the explicit representation of all information. The complete

description is accomplished with pointers that indicate, for each non-zero entry, the positions in a database where the information about the associated building-blocks (structure and parameters) is stored. Fig. 3 shows a representation example. Fig. 3(a) presents an evolved circuit with four building-blocks (b_1, b_2, b_3, and b_4). Fig. 3(b) shows the reduced matrix representation of the circuit. It carries the information about the building-blocks locations, but for simplicity purpose we use hereinafter the reduced matrix representation shown in Fig. 3(c).

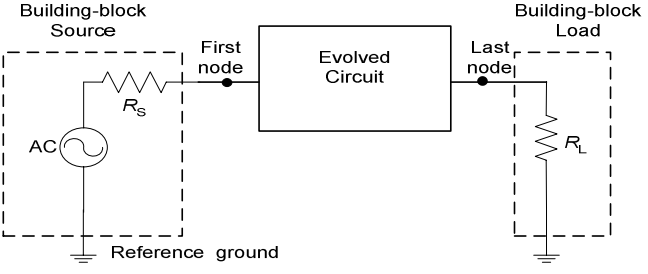


Fig. 2. Template circuit. The evolvable circuit is located between the first and the last node. The source and load are not changed in the evolution process.

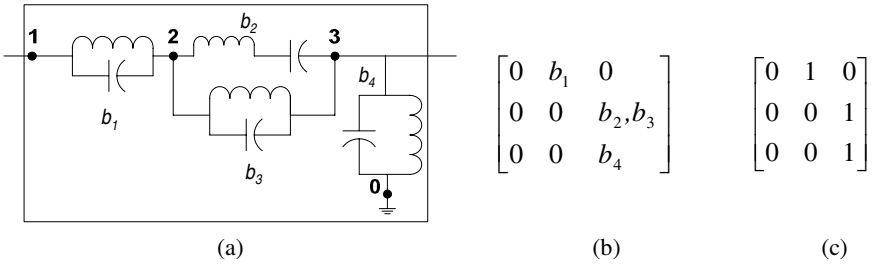


Fig. 3. Circuit representation (a) Evolved circuit (b)-(c) Location of building-blocks into the reduced matrix

Evaluation functions. Two objective-functions are defined: (1) the circuit performance, which is evaluated through a circuit simulator, and (2) the circuit size, given by the number of nodes in this work. The circuit simulator computes the frequencies responses (the gain) over a set of frequencies, defined by the user. After that, the algorithm calculates the absolute deviations average between the computed responses and the desired responses (an error measurement), which are predefined through a user-defined gain mask.

Initialization. The population is initialized with circuits (individuals) composed by building-blocks (genes), randomly selected from the database, connected via coupling rules. Only one building-block is placed at each position and each parameter of the

block is set up with a value randomly chosen from a user-defined range. It is possible to make use of rules defined by an expert user ([8]-[10] show examples of coupling schemes). Intelligent inputs greatly improve the quality of the initially generated circuits. In the case the expert inputs are not at hand, the algorithm runs a standard procedure, namely, it generates an arbitrarily sized matrix associated with a fully valid connected circuit. The second diagonal (above the main diagonal of the reduced matrix) is completely filled with randomly chosen building-blocks and each position of the other diagonals is randomly filled with high probability.

Classification and Selection. In order to generate high-performance but small circuits, a multi-objective selection approach – the crowded-comparison operator, extracted from the NSGA-II [13] – is applied in this method. The two evaluation functions (objective-functions) previously stated in this paper are taken into account, as in Fig. 4. The elite individuals of the population are found out by applying this classification method. This approach provides the balance between performance and size of the solutions, and, consequently, it becomes possible to naturally reduce the tendency of the process of producing larger circuits as the population evolves. Additionally, it allows the building-blocks identification, derived from the evolution process, which can be used in the next stage to produce competitive circuits with some degrees of structural redundancy (building-blocks naturally generated by the classification/selection mechanism). The selection scheme used in this work is the well-known binary tournament.

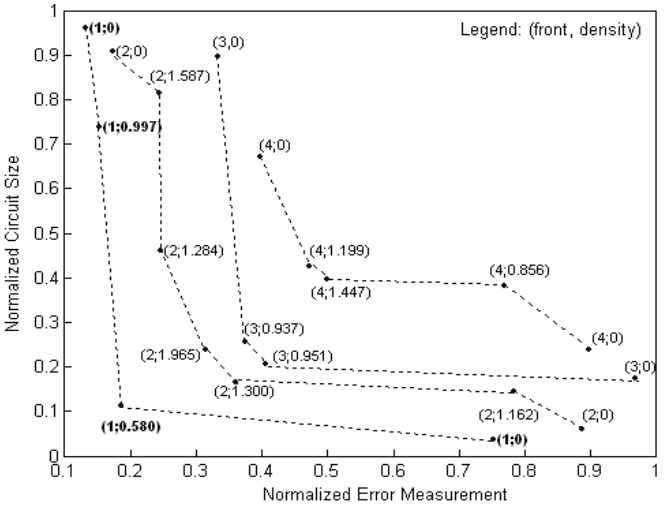


Fig. 4. Pareto Fronts of the minimization problem: In this case the first front (non dominated solutions) has four individuals – they achieved better fitness values than all the remaining population solutions

Crossover Operator. A new crossover operator is proposed. Fig. 5. sketches this operator. The crossover occurs as follows. Given two parent circuits represented by their reduced matrices, a cut point in parent matrix 1 is randomly chosen, such that four regions R_1, R_2, R_3 e R_4 are defined, as shown in Fig 5(a). After that, a region R_5 in parent matrix 2 is arbitrarily defined, as shown in Fig. 5(b). Figs. 5 (c)-(e) illustrate the offspring composition. Three types are possible and equiprobable: overlapping, as in Fig. 5(c); truncating, as in Fig. 5(d); and splitting configuration, as in Fig. 5(e). In the overlapping configuration, the R_1, R_2, R_3 e R_4 in the offspring matrix are from the parent matrix 1, the region R_5 is from the parent matrix 2, and the region R_6, is randomly chosen from parent matrix 1 or 2. In the truncating configuration, the region R_1 (from the parent matrix 1) is cascaded with the region R_5 (from the parent matrix 2). In the splitting configuration, the region R_5 from the parent matrix 2 is cascaded between the regions R_1 e R_4 after the splitting parent matrix 1. The regions R_6 in Figs. 5(c)-(e) represent a node set connected during the cascading operations (overlapping regions from the two parent matrices). Then, the proposed crossover operator can explore the containing knowledge in the parents, and also can promote the diversity of structures.

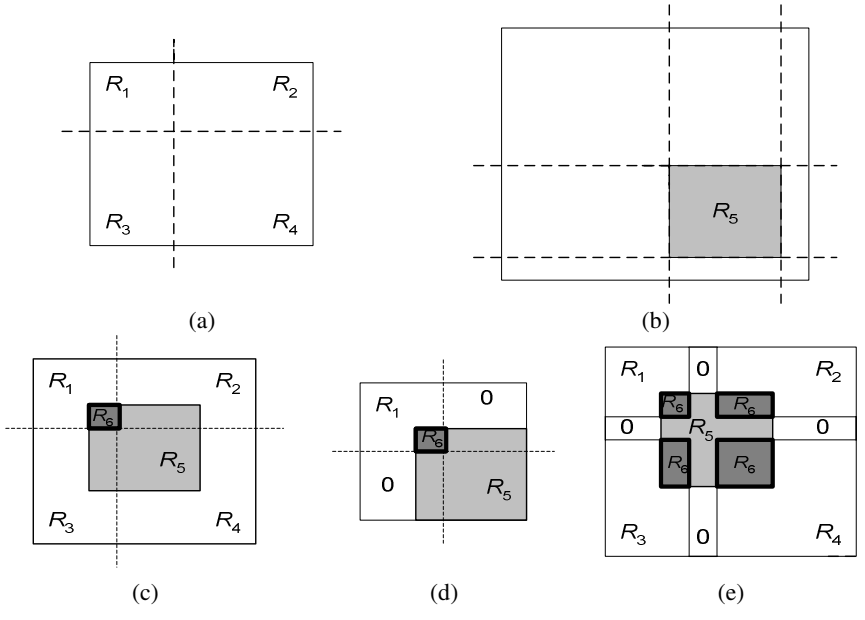


Fig. 5. Crossover Operator. (a) Parent matrix 1. (b) Parent matrix 2. (c) The offspring matrix after overlapping. (d) The offspring matrix after truncating. (e) The offspring after splitting.

Topology Mutation Operator. In this work, we propose four types of topology mutation. They are equiprobable. The circuit mutation is performed via one of the

following operations: (1) adding a randomly chosen building-block, without position restriction; (2) deleting a building-block, since the circuit remains connected; (3) deleting a node, by removing a row/column associated to the node, given that the circuit remains connected; (4) inserting a node, by adding a row/column associated to the new node, and a building-block in order to keep the circuit connected.

Parameter Mutation Operator. All the parameters of the building-blocks of the circuit may possibly suffer mutation. If a parameter is to be mutated, a new parameter value is randomly generated, observing a predefined range of possible values.

Local Search. In this memetic method, a local search process assists the Evolutionary Algorithm. It aims the fitness improvement of candidate circuits, slightly refining their parameters in order to avoid good topologies with non-optimized parameters values to be prematurely discarded. This process takes place in two points of the evolution cycle. After the classification process, the local search method is applied to each non-dominated individuals. Also, the local search procedure is carried out after the crossover/mutation procedure. Doing so, the topology space is explored and, subsequently, the parameters of the new topologies are improved. As a result, offspring solutions will be able to fairly compete with the current elite set for composing the elite of the next generation. We use the Simulated Annealing technique [15] with few iterations and predefined temperature values such that only a low computational effort is spent with each local search.

3 Experiment and Results

We have synthesized several different filters with several complexity levels using the proposed method. It successfully produced filters that complied with the desired specifications and the time spent for the entire synthesis process was modest. In this work, we chose the Nielsen filter due to its highly asymmetric bandpass response, which offers a considerable difficulty degree. This is a classical problem reported in previous works (see [2]-[4], and [16], for example), which allowed us to make comparisons. The specifications are presented in [3]: the passband is [31.2, 55.6] kHz; the lower stopband edge is at 20.0 kHz; the desired lower stopband gain is lower than −38 dB; the upper stopband is at [69.6, 85.0] kHz; the desired upper stopband gain is lower than −73 dB; and the desired gain above the stopband is lower than −55 dB. An inductor and a capacitor were defined as building-blocks, and the standard coupling scheme rules were used for initialization.

We used crossover probability of 100%, topology mutation probability of 20%, and parameters mutation probability of 5%. The circuit performance was observed at 100 frequencies. In all of the 10 runs we achieved results that accomplish the specifications. The best solution was a 4-nodes circuit (12 components). The solution was achieved after 41 generations with 30 individuals in the population (about 16,000

circuit evaluations). The results are shown in Figs. 6-8. We can notice in Fig. 6 that the size of the best circuit varies along the generations. The circuit changes their number of elements until it adapts to the specifications. It can be noticed in Fig. 7 that the proposed method naturally identifies and makes use of genetic building-blocks – for example, the parallel capacitor/inductor sub-circuits – created along the evolution process, as desired when we consider the building-block hypothesis [14]. The topology and the frequency response of the best circuit are shown in Fig. 8. Koza et. all. [2], using the Genetic Programming approach with 640,000 individuals in the population and after 199 generations (about 127,360,000 circuit evaluations), obtained a circuit with 38 components. Clearly, it is not a cost-effective design process. Grimblebly [3], using a hybrid genetic algorithm, achieved a 4-nodes circuit, but he did not mention the spent computational effort. Shin et. all. [4], using a multi-stage GA, obtained a circuit with 2,000 individuals in the population and after 400 generations (about 800,000 circuit evaluations) that complied with specifications, and did not present the structure of the circuit.

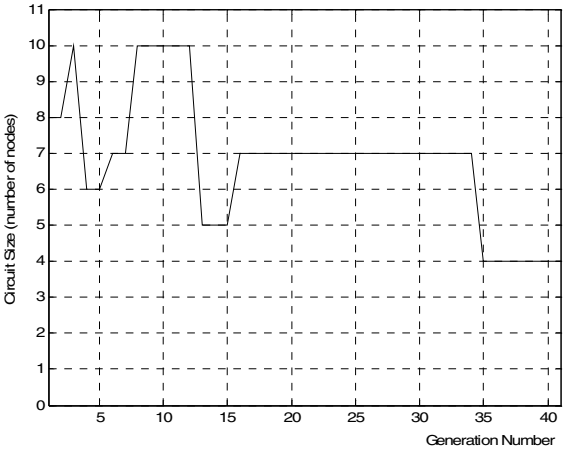


Fig. 6. Size evolution (number of nodes) of the best solution

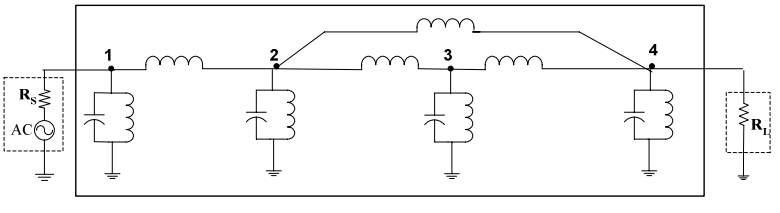


Fig. 7. The best topology to Nielsen filter experiment obtained in 10 runs

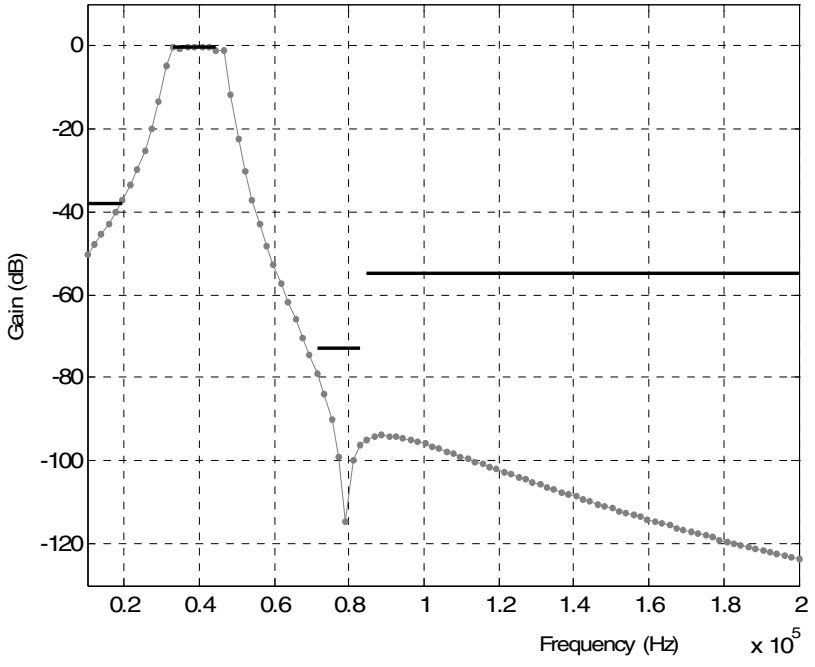


Fig. 8. The frequency responses of the best solution to Nielsen filter experiment. Legend: The thick black line represents the user-defined gain mask.

4 Conclusions

In this work, we proposed a Multi-objective Memetic Algorithm applied to the automated synthesis of the analog circuits. It is capable of generating circuits that meet the design specifications and present desirable characteristics such as structural regularity and controlled complexity. The expert knowledge-based approach allows the representation of a rich variety of circuit topologies. The system requires a little expert knowledge from user and has been successfully used to produce different filters with several complexity levels and the number of required circuit evaluations is modest, as shown to Nielsen filter experiment. The convergence process is very fast if compared to the evolutionary synthesis methods reported in the literature, and the method has the advantage of reducing troublesome trials to specify the design parameters in the conventional design procedure. We aim to introduce new design rules extending the representation for multi-port linear/nonlinear building-blocks. After that, it will be possible, in next works, to present an exhaustive study about new topologies considering more stringent specifications demanded in modern applications.

References

1. Hu, J., Goodman, E. D, and Rosenberg, R.: Robust and Efficient Genetic Algorithms with Hierarchical Niching and a Sustainable Evolutionary Computation Model. Proc. of the Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science, Springer, GECCO-2005, Seattle, Part I, (2005) 1220-1232
2. Koza, J. R., Bennett III, Andre, F.H., and Keane, M.A.: Automated Design for Both Topology and Components Values of Electrical Circuits using Genetic Programming. Genetic Programming 1996: Proceedings of the First Annual Conference. Stanford University, Cambridge MA. (1996) 123-131
3. Grimbleby, J. B.: Automatic Analogue Circuit Synthesis using Genetic Algorithms. IEEE Proceedings: Circuits, Devices and Systems, Vol. 147, No. 6. (2000) 319-323
4. Shin, A., and Histoshi, I.: Variable length Chromosomes for Analog Evolvable Hardware. Eds. A Ghosh and S Tsutsui, Advances in Evolutionary Computation, Springer-Verlag, (2003) 643-662
5. Krasnogor, N., and Smith, J.: A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues. IEEE Transactions on Evolutionary Computation, Vol. 9, No. 5, (2005) 575-588
6. Dastidar, T. R., Chakrabarti, P.P., and Ray, P.: A Synthesis System for Analog Circuit Based on Evolutionary Search and Topological Reuse. IEEE Transactions on Evolutionary Computation, Vol. 09, No. 2, April (2005) 211-225
7. Levy, R., Snyder, R. V., and Matthaei, G.: Design of Microwave Filters. IEEE Transactions on Microwave Theory and techniques, Vol. 50, No. 3, March (2002) 783-793
8. Rosemberg, U., and Amari, S.: Novel Coupling Schemes for Microwave Resonator Filters. IEEE Transactions on Microwave Theory and Techniques, Vol 50, No. 12, December (2002) 2896-2902
9. Yildirim, N., Sen, O. A., Sen, Y., Karaaslan, M., and Pelz, D.: A Revision of Cascade Synthesis Theory Covering Cross-Coupled Filters. IEEE Transactions on Microwave Theory and Techniques, Vol. 50, No. 06, June (2002) 1536-1553
10. Amari, S., and Rosenberg, U.: Direct Synthesis of a New Class of Bandstop Filters. IEEE Transactions on Microwave Theory and Techniques, Vol 52, No. 02, February (2005) 607-616
11. Mesquita, A., Salazarand F. A., and Canazio, P. P.: Chromosome Representation through Adjacency Matrix in Evolutionary Circuit Synthesis. NASA/Conference on Evolvable Hardware. (2002) 102-112
12. Im, C.-H., Jung, H.-K., and Kim, Y.-J.: Hybrid Genetic Algorithm for Electromagnetic Topology Optimization, IEEE Transactions on Magnetic, Vol. 39, No. 5, September (2003) 2163-2169
13. Deb K., Pratap A., Agarwl, S., and Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, April (2002) 182-197
14. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley (1989)
15. Michalewicz, Z., and Fogel, D.B.: How to Solve It: Modern Heuristics. Spring Verlag, Berlin Heidelberg New York (2002)
16. Nielsen, I.R.: A C-T Filter Compiler –From Specification to layout. Analog Integrated Circuits and Signal Processing. Vol 7. (1995) 21-33

A Hybrid Learning Strategy for Discovery of Policies of Action

Richardson Ribeiro, Fabrício Enembreck, and Alessandro L. Koerich

Programa de Pós-Graduação em Informática Aplicada (PPGIA)
Pontifícia Universidade Católica do Paraná
Rua Imaculada Conceição, 1155, CEP 80215-901
Curitiba, Paraná – Brasil
{richard, alekoe, fabricio}@ppgia.pucpr.br

Abstract. This paper presents a novel hybrid learning method and performance evaluation methodology for adaptive autonomous agents. Measuring the performance of a learning agent is not a trivial task and generally requires long simulations as well as knowledge about the domain. A generic evaluation methodology has been developed to precisely evaluate the performance of policy estimation techniques. This methodology has been integrated into a hybrid learning algorithm which aim is to decrease the learning time and the amount of errors of an adaptive agent. The hybrid learning method namely K-learning, integrates the Q-learning and K Nearest-Neighbors algorithm. Experiments show that the K-learning algorithm surpasses the Q-learning algorithm in terms of convergence speed to a good policy.

1 Introduction

Reinforcement Learning (RL) is a computational paradigm of learning where an algorithm attempts to maximize a measure of performance based on the reinforcements (reward or punishment) that it receives when interacting with an unknown environment [11]. In the literature one can find several approaches to RL and examples of different applications using intelligent agents [8] [10] [11] [14] [18] [19].

In an attempt to optimize the learning task of an agent, many authors have integrated different learning methods, creating the so-called hybrid learning methods, with the aim that one can compensate the weaknesses of each other, leading to more robust agents with higher tolerance to faults [4] [7] [15] [17]. However, the evaluation of the action policies produced by these hybrid methods is a complex task due to the lack of generic mechanisms that allow measuring the performance of a learning agent without relaying on the knowledge of the problem domain (or independently of the problem domain). Current evaluation mechanisms are not generic enough to evaluate action policies in different environments.

In this paper we propose a novel hybrid learning method, unifying RL and instance-based learning algorithms. The aim of such a hybrid method is to discover good action policies in a short timeframe. Furthermore, we propose a novel generic methodology to evaluate the quality of the policy discovered by RL-based adaptive

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 268 – 277, 2006.

autonomous agents. This methodology allows the estimation of agents' performance without relaying on the knowledge of the domain and in short simulation periods. In the proposed methodology, the measure of performance of an agent is proportional to the number of correct decisions produced by its action policy in a given environment. A policy represents a space of states with initial and target states where the transition between states is determined by the actions. Therefore, a correct decision is reached when the agent finds a way to reduce the cost between the initial and target states. When this occurs for all candidate states, an optimal policy was discovered. The proposed evaluation methodology allows observing the behavior of the learning algorithm as a function of the number of iterations, configuration of environment and values of parameters related to the algorithm. The development of a hybrid learning mechanism was possible because of these above mentioned characteristics of the evaluation methodology.

This article is organized as follows: Section 2 presents some hybrid learning methods. Section 3 describes the evaluation mechanisms employed in different classes of problems. The novel performance evaluation methodology as well as an overview of the Q-learning, the K-NN, and the K-learning algorithms are presented in Section 4. Experimental results are presented in Section 5. Finally, in the last section some conclusions are drawn as well as perspectives for future research.

2 Hybrid Learning Methods

Different hybrid learning methods have been proposed in the literature with the aim of integrating different learning methods. Henderson et al. [7] proposes a hybrid model that combines RL with supervised learning. RL is used to optimize the average of rewards between the communication of systems that use huge data sets and that also have large state spaces. The supervised learning is used to constrain the learned policy of one part of state's space and modeling a policy with the data used in a given moment. Downing [4] proposes a hybrid technique that combines RL with trees based on genetic programming. This technique adds a new element to a set of functions of the genetic programming and produces a system, whose actions are strengthened by genetic programming so that the successive functioning of some trees shows the improvement of task adjustment performance. Figueiredo et al. [6] propose a hybrid neuro-fuzzy model based on RL. This model presents important characteristics as: automatically learning the model structure; self-adjusting to the parameter performance associated with the structure; and capacity to learn actions that must be taken when the agent is in a state of the environment. RL is used in this hybrid model to determine more properly actions to be executed for a given state. Rayan [15] proposes a system that incorporates techniques of symbolic planning with RL with the aim of producing a system capable of intensifying each method. The system uses a new behavior representation, which is defined in terms of desired consequences, but letting the implementation of policy to be learned by RL.

We use some concepts presented in such works to develop a hybrid method, namely *K-learning,* that integrates two well-know learning techniques used in adaptive autonomous agents: RL (Q-learning algorithm) and Instances-Based Learning (K-NN algorithm). This hybrid method emphasizes the strengths of each learning

algorithm: the K-NN allows that states with similar characteristics have similar rewards, anticipating rewards and decreasing the number of iterations to the Q-learning. On the other hand, the Q-learning guarantees that an optimal policy be found along the iterations.

3 Policy Evaluation Mechanisms

Measuring performance of a learning agent is not a trivial task. It can be measured according to its proposed task and mainly according the desired objective. For such an aim, some measures such as speed of convergence for the optimal or almost-optimal behavior, total reinforcements obtained by the agent and percentage of optimality have been used [8].

Ernst et al. [5] present a metric of quality of solution produced by an agent, as a mechanism to evaluate the performance of RL algorithms. A stationary policy is used to calculate the expected values which are further compared with regression algorithms on a set of examples. These examples have initial states chosen from a set of vectors generated by the Q-learning algorithm and the average expected value of the stationary policy is computed. Ramon [12] used decision trees to find an approximated function Q to guarantee the convergence to an optimal policy. He proposes a learning algorithm that generates a new estimate Q' from a preceding estimate Q. The author uses three types of measures to evaluate the performance of an algorithm: an amount $Q'(s,a)$ up to date that is wanted; an estimate $\tilde{q}(s,a)$ of Q' that is measured in exploration executed by algorithm; a measure $\tilde{q}(c)$ where $c \subseteq SXA$ is an average of function $\tilde{q}$ on an abstraction c, where S is the set of states and A the set of action. The algorithm stores in memory the values of the third metric of form to generate statistics. Lev et al. [9] use heuristic value function that evaluates the performance of an agent by computing the states foreseen for $\tilde{V}^*$ and selects an operator who leads to the most promising state being $\tilde{V}^*$ the optimal value of the function.

Generally, the evaluation of policies is not flexible enough to deal with different kind of problems. Moreover, they require too much knowledge on the domain to define heuristics related to the problem solution. Due to this fact, there is a lack of generic evaluation mechanisms that can be used in problems that involve RL.

4 A Novel Evaluation Methodology

The novel evaluation methodology analyzes the performance of RL algorithms assuming that there is a reward function capable of reproducing an optimal action policy. We consider that the performance of an agent is proportional to the number of correct decisions produced by its action policy in a given environment. A policy represents state space with initial and target states, where the transition between states is determined by actions. Therefore, a correct decision is attained when the agent finds the path between the initial and target state which has the lowest cost. Otherwise, an error is found. When this occurs for all candidate states, it can be stated that an *optimal policy* was discovered. In this way, the evaluation of a policy is carried out

through a problem solving algorithm capable of finding the optimal path between two states where the heuristic function corresponds to the values returned by the reward function. Moreover, the amount of states visited can also be taken into account.

We use the *A Star* algorithm (*A**) to find the best solution which is complete and optimal [14]. Since the *A** produces the best policy for a given heuristic, the quality of the policy discovered by different learning algorithms can be stated through the comparison of the policy found by them against the policy produced by the *A**. This methodology shows the proximity between the current policy of the agent and the optimal policy. A correct decision occurs when the agent is able to reach the target with an optimal cost (shortest path). We use the number of states and the additional cost associated with each state (value of reward function) to evaluate the total cost of a path.

To evaluate the performance of learning algorithms, we place the agent in an unknown environment and adjust the parameters for each possible initial state. Further, we set the learning parameters of the agent. Thus, it is possible to analyze the performance of algorithms through the learning curves.

```
Algorithm EvaluateAlgoritm_X(PQ,PX)//PQ is action policy of the
Q-learning algorithm and PX is action policy of the X algorithm;
1 For a given environment PQ=0, PX=0;
2 For each iteration of the Q-learning repeat:
      Efficiency_Q_Learning ← evaluatePolicy(PQ)
         If stop_condition() = false then
             Return to 2
         Else
             Go to step 3
3 For each state s learned by the Q-learning repeat:
             PX ← generatePolicy(PQ)
4 Efficiency_X ← evaluatePolicy(PX);
5 Go to step 2 if continuing the learning of the Q-learning;
6 Return (PQ,PX);
7 End.
```

Fig. 1. Pseudo code of the performance evaluation algorithm

```
Algorithm EvaluatePolicy(P);
1 Initiating Correct=0, Wrong=0, CostP=0, CostA*=0;
2 For each s ∈ S:
     CostP = cost(s, s_goal, P);
     CostA*= cost(s, s_goal, PA*);
     If CostP <= CostA*
         Correct ← Correct+1;
     Else
         Wrong ← Wrong+1;
          3 P ← (Correct / (Correct + Wrong)) * 100;
4 Return (P);
5 End.
```

Fig. 2. Pseudo code of the algorithm of evaluation of a policy *P*

Fig. 1 and 2 present the pseudo code of the performance evaluation algorithm which compares the performance of the Q-leaning algorithm [20] and an algorithm *X* considering any estimation policy. Fig.1 shows that in each iteration is applied an evaluation function that computes the efficiency of the algorithms. Each iteration represents the state transition by applying an action *a* using the current policy. In step 3 the X algorithm is used to generate a new policy derived from *PQ*. Finally, the efficiency of *PX* is computed and the process continues until a stop condition is reached. Fig. 2 presents the algorithm used for measuring the efficiency of a given policy *P*. The *cost* function is used to find the best path between states *s* and s_goal from a given policies

4.1 Using the K-NN Algorithm to Generate Policies

The K-NN algorithm [1] is used to generate the values in the learning table of an agent based on the Q-learning algorithm. The K-NN algorithm receives as input a set of instances produced by a policy which is generated during learning phase. The aim is to generate a new set of values in an attempt to approximate the agent to an optimal policy.

We define as *arrangement* the set of instances generated during a cycle of steps carried out by the Q-learning algorithm. It is interesting to notice that for each state, four instances are generated (one for each action) and they represent the values learned by the agent. Therefore, the number of instances is the same for any policy. Thus, each instance of the arrangement has attributes for representation of the state in form of the expected reward for the actions north, south, west and east; an action and an attribute representing the reward for the action.

These vectors generated by the Q-learning agent during learning will be compared with a similarity function (cosine) and after that, the nearest instances will be found. Eq. 2 describes how the K-NN can be used to generate estimates of the values for the arrangement formed by the training instances.

$$PKNN\ (s,a) \leftarrow \frac{\sum_{i=1}^{K} PQ_i(.,.)}{K} \tag{1}$$

where *PKNN(s,a)* represents the values of the reward calculated for a given state *s* and an action *a*, *K* represents the number of neighbors used and $PQ_i(.,.)$ represents the 1st-nearest neighbor instance found in training set generated by the policy *PQ*.

4.2 The K-Learning Method

The method described previously can generate intermediate policies; however, it is not guaranteed that these policies present a good quality. In some cases, the values of rewards for a given state change closer to a set of correct rewards. On the other hand, states that have high rewards can become less important. This occurs because all the states will have their values of reward modified, even those whose rewards are good.

The *K-learning* is a hybrid learning method resulting from the merge of the Q-learning and K-NN algorithms which attempts to solve the above mentioned problem by selecting always the policy that improves the performance of the agent. The learning of

the Q-learning algorithm is stored in a table of name *PQ(s,a)* and the learning of the K-NN in a table *PK(s,a)*. The best values of these tables obtained during the learning phase are stored in a new table, which is called *PKI(s,a)* (this is the policy generated by K-learning).

This method has been created from experimental observations where the efficiency of the methods varies during the learning period. In many cases, the performance of the K-NN has made the rewards discovered for the Q-learning less interesting, i.e., states that produce correct decision started to produce errors[1]. To solve this problem, the K-learning modifies only rewards of states that lead to learning errors. When one of the methods is superior to the other, the agent switches the learning method. This makes the agent able to find the action policy that maximizes its performance and decreases the amount of necessary steps. Fig. 3 presents the pseudo code of the K-learning method.

```
Algorithm K_learning(PQ,PKI);
1 Initiating PQ=0, PKI=0;
2 For each iteration repeat:
     While Stop_Conditions <> true do
      If evaluatePolicy(PQ) > evaluatePolicy(PKl)
           Then
                CurrentPolicy ← PQ
           Else
               CurrentPolicy ← PKl
3 Return (PQ,PKI);
4 End.
```

Fig. 3. Pseudo code of the K-learning method

Fig. 3 shows that the performance of the Q-learning and K-NN algorithms is compared for all iteration. When the performance of one of them is higher, the policy is stored into a new table (*CurrentPolicy*) which represents the current action policy. The results of this hybrid learning method are presented in Section 5.

4.3 The Simulated Environment

To represent the environment of simulation (road mesh), the crossing had been represented as states. The states of the virtual environment have characteristics of situation traffic: free, low congestion, high congestion, blocked and unknown. Each situation has different values that are used to compute the rewards received by the agent. The agent moves taking one of the following actions: forward, backward, right and left. After each move (transition), the agent will start to have a new state generated by the environment and new actions could be taken. The agent will know if its action was positive or negative through the reward granted by the environment. Eq. 3 describes the current state *s* updated by the next state added to one reward *r*.

$$Q(s,a) \leftarrow \partial(s',a') + r \tag{2}$$

[1] The error definition has been discussed in Section 4.

The transition value ∂ is computed by the agent for a reinforcement scale signal. The agent must choose actions that tend to increase the sum of the values of the reinforcement signal r, as time goes by. The agent can learn how to make this through systematic try-and-test while it interacts with the environment. Thus, the agent will know if its action was positive or negative through the change of behavior of the environment (traffic). If the agent finds itself in a congested traffic state and after this its action goes into to the little congested state, it then receives a positive reward, but if the action takes it to a very congested state a negative reward is received.

5 Experiments

Experiments were carried out with the Q-learning algorithm to evaluate its efficiency considering factors such as: the number of iterations necessary for the agent to reach its best efficiency; the quality of reward policy; variations in learning rate; discounting factor and the values of reward for the congestion conditions. In this section we present experiments that allow identifying the importance of each one these factors in the learning process.

The experimental protocol to evaluate the efficiency of the algorithms includes twenty repetitions of each type of experiment using ten different environments sizes. This is required due to the variation in the algorithm efficiency in single environments. This occurs because the values generated during the learning phase are stochastic. The values of efficiency presented in this section represent average values. The best values used as learning rate for the Q-learning algorithm are between 0.10 and 0.20 and the best values for the discount factor are between 0.85 and 0.95.

Fig. 4 presents a comparative among: efficiency, number of states and amount of steps. Fig. 4 shows that for environments smaller than 25 states, about 500 steps are necessary to reach the best efficiency. For environments up to 81 states we observe that it is necessary about 5,000 steps. Environments above 100 states need a more than 20,000 steps to reach the best efficiency. Fig. 4 and 5 show that in large environments the agent needs a high number of cycles due to the need of visiting each state many times to accomplish the learning.

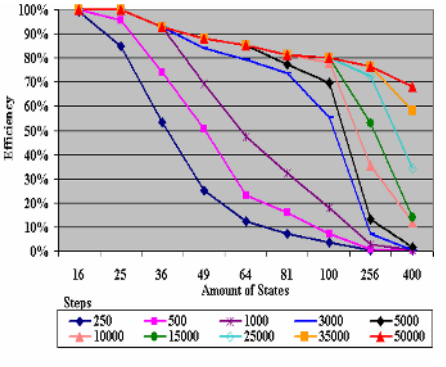


Fig. 4. Learning curves for the Q-learning: efficiency *vs* n. of states *vs* n. of cycles

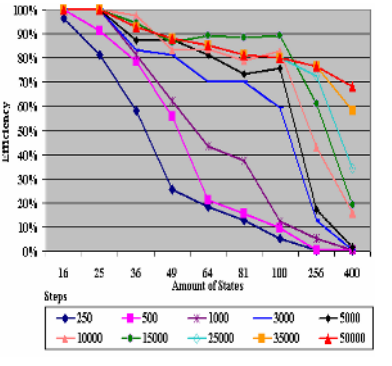


Fig. 5. Leaning curves for the 1-NN: efficiency *vs* n. of states *vs* n. of cycles

The instances generated by the Q-learning and used by K-NN are stored into a K-Table. Further, the policy represented in the K-Table is evaluated using the algorithm *A** as discussed in section 4. The experiments with the K-NN algorithm have taken place in environments with size up to 400 states and with up to 50,000 cycles. The average efficiency of the K-NN algorithm was evaluated considering K=1 because this was the value that has presented the best results.

It is possible to observe that the K-NN algorithm presents a better performance than Q-learning for most of the environment sizes (Fig. 4 and 5). This fact motivates us to propose the hybrid learning method, namely K-learning.

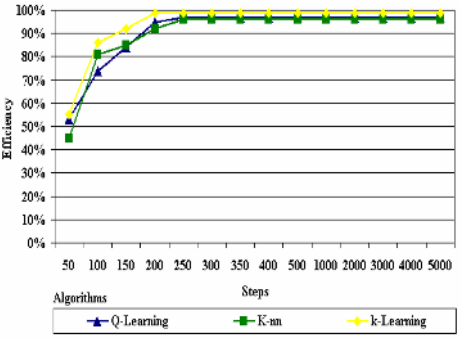


Fig. 6. Efficiency of the Q-learning, 1-NN and K-learning algorithms for a 16-state environment

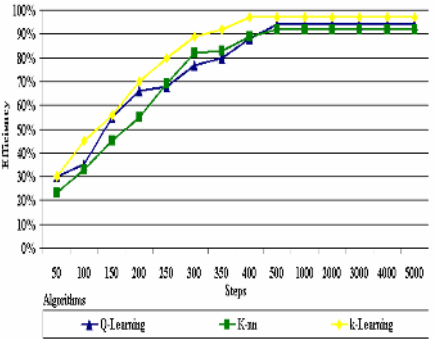


Fig. 7. Efficiency of the Q-learning, 1-NN and K-learning algorithms for a 25-state environment

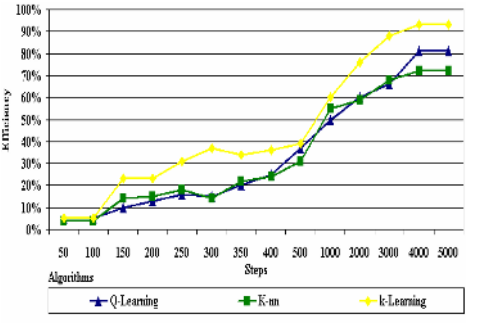


Fig. 8. Efficiency of the Q-learning, 1-NN and K-learning algorithms for a 64-state environment

5.1 K-Learning Versus Q-Learning and K-NN

To evaluate the performance of the K-leaning algorithm we have carried out experiments in environments with size between 16 and 64 states. Fig. 6 to 8 show the efficiency curves for the K-learning hybrid method, the Q-learning and the K-NN algorithms. The curves demonstrate that, in general, the K-learning hybrid method presents a better efficiency than Q-learning and K-NN algorithms in any learning

cycle. We can also observe that the number of steps necessary to find a good action policy decreases significantly relative to the Q-learning and K-NN algorithms. In 16-state environments there is an average reduction of 18% in the number of steps to achieve the best efficiency. For 25-state environments the reduction is 20%. Finally, for 64-state environments the K-learning method has achieved the best efficiency with 12% less steps. The values of the Tab. 1 present the superiority of the K-learning method to the Q-learning and K-NN algorithms. It is possible to observe that the K-learning method has a higher average performance relative to the other methods. It also requires a lower number of iterations to find a good action policy. This shows that this method is able to adapt itself to different environments. The good average performance results from the policy generation technique gets the values learned by the agent closer to the values of an optimal action policy.

Table 1. Average superiority of the K-learning method to the Q-learning and K-NN algorithms

Environment Size (number of states)	Q-learning (%)	K-NN (%)
16	2	3
25	3	5
64	12	21

6 Conclusion and Discussion

This paper presented a novel hybrid learning method that is more efficient that conventional learning algorithm such as the Q-learning, K-NN algorithms. The experiments carried out on different environment sizes have show that even having a higher computational cost, the K-NN algorithm achieves satisfactory results since it generally finds superior or equivalent solutions to Q-learning with a low number of iterations. The hybrid K-learning method has integrated the robustness of the Q-learning with the efficiency of the K-NN algorithm to modify the learning values. This hybrid learning method has succeeded to optimize the agent learning in terms of efficiency and the number of iterations necessary to achieve a good performance.

In this paper we have also presented a novel evaluation methodology which is robust in dealing with partially known and complex environments. Such an evaluation methodology also allows the setup of suitable learning parameters for RL algorithms. This is an important point because these factors have great relevance in the performance of a learning agent.

Even though the results are encouraging, additional experiments are necessary to answer some open questions, such as: (i) the comparison of different forms of updating the current policy. The update of the current policy could be partial rather than global (only with the best reward values); (ii) a heuristic function could be used to speedup the RL [3]; (iii) A multi-agent system could be used for exploring distant regions of the target state where rewards are low. Multi-agent reinforcement learning is a very active subject of research [2] [16]; (iv) another question consists in evaluating the algorithms in dynamic and noisy environments. Since the K-learning uses results from different algorithms, it should be robust in situations where the reward

values vary since is well known that the K-NN is robust to deal with noise in training data; however, this hypothesis will be verified in future research.

References

1. Aha, D.W., Kibler, D., Albert, M.K. Instance-based Learning Algorithms, Machine Learning. 6(1), pp. 37-66, 1991.
2. Almeida, A., Ramalho, G. L., Santana, H. P., Tedesco, P., Menezes, T. R., Corruble, V., Chevaleyre, Y. (2004). Recent Advances on Multi-Agent Patrolling. In Advances in Artificial Intelligence – SBIA 2004. LNAI 3171, pp. 474-483. Berlin: Springer
3. Bianchi, R. A. C., Ribeiro, C. H. C., Costa, A. H. R. Heuristically Accelerated Q-learning: A New Approach to Speed Up Reinforcement Learning. XVII SBIA'04: pp: 245-254, 2004.
4. Downing, K. L. Reinforced Genetic Programming. Genetic Programming and Evolvable Machines, 2(3):259-288, 2001.
5. Ernst, D., Geurts, P., Wehenke, L. Tree-Based Batch Mode Reinforcement learning. Journal of Machine Learning Research, April 2005, Volume 6, pp 503-556
6. Figueiredo, K., Vellasco, M., Pacheco, M., Souza, M., Reinforcement Learning Hierarchical Neuro-Fuzzy Politree Model for Control of Autonomous Agents," his, pp. 130-135, Fourth Int. Conference on Hybrid Intelligent Systems (HIS'04), 2004.
7. Henderson, J., Lemon, O., Georgila, K.. Hybrid reinforcement/supervised learning for dialogue policies from COMMUNICATOR data. In Proc. IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems, Edinburgh, 2005.
8. Kaelbling, L. P., Littman, M. L, Moore, A. W., Reinforcement Learning: A survey. Journal of Artificial Intelligence Research. Vol. 4, pp. 237-285, 1996.
9. Levner, I., Bulitko,V., Madani, O. , Greiner, R.. Performance of Lookahead Control Policies in the Face of Abstractions and Approximations. Technical Report. Publisher: Springer-Verlag., Vol. 2371/2002. pp. 299-307.Maes, P. Artificial Life Meets Entertainment: Lifelike Autonomous Agents, Communications of ACM, Vol. 38, No. 11, pp. 108-114, 1995.
11. Mitchell, T. Machine Learning. Boston:McGraw-Hill, 1997.
12. Ramon, J. On the convergence of reinforcement learning using a decision tree learner. Proceedings of ICML-2005 Workshop on Rich Representation for Reinforcement Learning, Bonn, Germany, 2005.
13. Ribeiro, C. H. C. A Tutorial on Reinforcement Learning Techniques, Int. Joint Conference on Neuronal Networks, Washington: INNS Press, 1999.
14. Russel, S., Norvig, P., Inteligência Artificial, 2 ed., Rio de Janeiro: Editora Elsevier, 2004.
15. Ryan, M. R. K. Hierarchical Reinforcement Learning: A Hybrid Approach. PhD Thesis, University of New South Wales, School of Computer Science and Engineering, 2004.
16. Santana, H., Ramalho, G., Corruble, V., Ratitch, B. Multi-Agent Patrolling with Reinforcement Learning In proc. 3rd International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'04). pp. 1122-1129. New York: ACM
17. Siedlecki, W., Sklansky, J., A note on Genetic Algorithms for Large-Scale Selection, Pattern Recognition Letters, Vol. 10, pp. 335-347, 1989.
18. Sutton, R.S., Barto, A.G. Reinforcement Learning: An Introduction. Cambridge, MA. MIT Press, 1998.
19. Tesauro, G. Temporal Difference Learning and TD-Gammon. Communications of the ACM, Vol. 38, No. 3, pp. 58-68, 1995.
20. Watkins, C. J. C. H., Dayan, P. Q-learning, Machine Learning, 8 ed., pp. 279-292, 1992.

A Fractal Dimension Based Filter Algorithm to Select Features for Supervised Learning

Huei Diana Lee[1,2], Maria Carolina Monard[1,2], and Feng Chung Wu[1,2]

[1] Laboratory of Computational Intelligence – LABIC,
University of São Paulo – USP,
Institute of Mathematics and Computer Science – ICMC,
P.O. Box 668, 13560-970 – São Carlos, SP, Brazil
[2] Bioinformatics Laboratory – LABI,
West Paraná State University – UNIOESTE, Itaipu Technological Park – PTI,
P.O. Box 1511, 85856-970 – Foz do Iguaçu, PR, Brazil
huei@unioeste.br, mcmonard@icmc.usp.br

Abstract. Feature selection plays an important role in machine learning and is often applied as a data pre-processing step. Its objective is to choose a subset from the original set of features that describes a data set, according to some importance criterion, by removing irrelevant and/or redundant features, as they may decrease data quality and reduce the comprehensibility of hypotheses induced by supervised learning algorithms. Most of the state-of-art feature selection algorithms mainly focus on finding relevant features. However, it has been shown that relevance alone is not sufficient to select important features. It is also important to deal with the problem of features redundancy. For the purpose of selecting features and discarding others, it is necessary to measure the features' goodness (importance), and many importance measures have been proposed. This work proposes a filter algorithm that decouples relevance and redundancy analysis, and introduces the use of Fractal Dimension to deal with redundant features. Empirical results on several data sets show that Fractal Dimension is an appropriate criterion to filter out redundant features for supervised learning.

1 Introduction

Supervised learning algorithms take as input a training set of N classified instances $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ for some unknown function $y = f(\mathbf{x})$, where the $\mathbf{x}_i$ values are typically vectors of the form $(x_{i1}, x_{i2}, ..., x_{iM})$, and x_{ij} denotes the value of the j-th feature (or attribute) X_j of $\mathbf{x}_i$. For classification purposes, the y values are drawn from a discrete set of N_{Cl} classes, *i.e.* $y \in \{C_1, C_2, ..., C_{N_{Cl}}\}$. From that training set, a learning algorithm induces a *classifier*, which is a hypothesis $\mathbf{h}$ about the true unknown function f. Given new $\mathbf{x}$ values, the classifier predicts the corresponding y values. Theoretically, a greater number M of features should provide a greater discriminating power. Nevertheless, with a limited amount of data this may not happen in the presence of irrelevant and/or redundant features, which frequently confuse the learning algorithm.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 278–288, 2006.

The objective of Feature Selection — FS — can be formalized as follows [1]: let $X' \subset X$ be a subset of features and $f'(\mathbf{x}')$ the value associated to instances described by features in X'. The objective of FS is to select a minimum feature subset X' such that $\mathbf{P}(C|y = f'(\mathbf{x}')) \approx \mathbf{P}(C|y = f(\mathbf{x}))$, where $\mathbf{P}(C|y = f'(\mathbf{x}')))$ and $\mathbf{P}(C|y = f(\mathbf{x}))$ are the probability distributions of the N_{C_l} possible classes given the feature values in X' and X, respectively. This minimum subset X' is named the *optimal* subset. In this context, the FS problem can be characterized under two aspects: how attributes are evaluated, and how the feature selection algorithm and the learning algorithm that use the selected attributes will interact.

Feature evaluation can be performed by feature selection algorithms in two main ways: individual evaluation and subset evaluation. On the one hand, individual evaluation is computationally less expensive, as this approach assesses individual features and assigns them weights (ranks) according to their degree of importance to the class. Nevertheless, this approach is incapable of detecting redundant features because these features are likely to have similar rankings. On the other hand, the subset evaluation approach can handle both feature relevance and feature redundancy. However, unlike individual evaluation, in this approach evaluation measures are defined against a subset of features, thus showing a high computational cost. Interaction between feature selection and learning algorithms can be widely categorized into *wrapper* and *filter* models [2,3]. Supervised wrapper models use the classifier itself to measure the performance of the feature subsets. Typically, they iteratively measure the quality of subsets by evaluating predictive accuracy of the class labels until the best feature subset is encountered or a stop criterion is met. As a consequence, wrappers present a high computational cost. On the other hand, filter models do not explicitly measure accuracy of predicting the class labels, they evaluate subsets or individual features related to their ability to determine the class labels. Apart from being independent from the learning algorithm, the filter model presents a lower computational cost than the wrapper model.

In this work, we propose a filter feature selection algorithm which decouples the analyses of relevance and redundancy, and considers the Fractal Dimension — FD — concept to treat redundancy of attributes. In spite of the frequent use of FD when detecting clusters and indexation of high dimensional structures, to the best of our knowledge, it has not been used to perform FS in the context of supervised learning, as proposed in this work. Experimental results using several data sets and FS algorithms often cited in the literature as well as the proposed algorithm, show that FD is an appropriate criterion to deal with feature redundance.

The remainder of this paper is organized as follows. Section 2 briefly describes the issue of importance of attributes. Section 3 introduces the fractal dimension concept and presents the proposed algorithm. Section 4 describes the empirical study of the proposed method comparing it with other representative FS methods. Section 5 concludes the paper and suggests some future work.

2 Importance of Attributes

In order to select attributes it is necessary to define what the meaning of a good attribute is, *i.e.* to answer the following question: *It is important related to what?* Furthermore, attribute's importance estimation should be applied to both, individual evaluation and subset evaluation, no matter the search strategy used. The evaluation issue is complex and multidimensional [2]. For example, evaluation can be considered in terms of: (1) improvement of the classifier accuracy, or (2) simplification of the constructed model to make it more comprehensible. In a general way, the importance of an attribute can be defined as:

Definition 1 *(Importance of an Attribute) [2]: An attribute is said to be important if its removal causes a degradation of the importance measure calculated from the remaining attributes.*

Different importance measures have been proposed, and they can be broadly divided into five categories: dependency, consistency, information, distance and classifier accuracy rate [2,4]. Some of these measures to evaluate attributes or to determine related to what they are considered important, are presented next.

Definition 2 *(Probabilistic Importance — Dependency Measure) [5]: An attribute X_j is said to be important if and only if there exists any x_{ij}, y and s_{ij} for which $P(X_j = x_{ij}, S_j = s_{ij}) > 0$ such that*

$$P(Y = y|X_j = x_{ij}, S_j = s_{ij}) \neq P(Y = y|S_j = s_{ij})$$

According to this definition, X_j is important if the class probability given all attributes can change with the elimination of knowledge about feature X_j where s_{ij} stands for the subset of features without x_{ij}. Another importance definition which enables to detect attribute redundancy considers the FD concept.

Definition 3 *(Fractal Dimension Importance — Dependency Measure) [6]: Given the FD calculated considering all attributes from a data set, an attribute is said to be important if its removal causes a significative*[1] *variation in the value of the recalculated FD.*

Some examples of applications that use fractal theory include determining indexing structures of high dimensionality and detection of clusters. Note that FD does not differentiate the class attribute from the other attributes. However, fractal theory is still not much used for feature selection whenever supervised learning algorithms are considered, as proposed in this work.

The FS problem can also be seen as the problem of finding an optimal subset for a specific learning algorithm, as shown by the next definition.

Definition 4 *(Accuracy Importance — Accuracy Measure) [3]: Given a learning algorithm $\mathcal{I}$ and a sample of data S with attributes X_1, X_2, ..., X_M and distribution $\mathcal{D}$ over a space of labeled examples, an optimal subset of features, X_{opt}, is the subset of features such that the induced classifier accuracy $\boldsymbol{h} = \mathcal{I}\ (\ \mathcal{D}\ )$ is maximum.*

[1] It depends on what the user establishes as a significative variation.

3 A Fractal Dimension-Based Filter Algorithm

Most feature selection algorithms which deal with both feature relevance and feature redundancy, evaluate features throughout subset evaluation. Although these methods usually present better results when compared with methods that do not consider the attributes redundancy problem, the high computational cost can make them inefficient for data sets with a large number of features. Recently, the use of the filter approach in a framework that decouples relevance and redundancy analyses was proposed — Figure 1 [1]. This framework presents the advantage that by detaching the analysis of relevance and the elimination of redundant features, the computational cost to search for a subset that approximates the optimal subset of features can be diminished. In this work, we

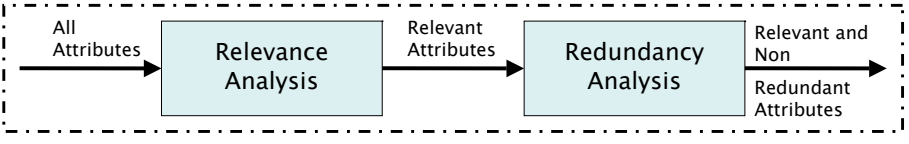


Fig. 1. A Framework for feature selection

propose an algorithm, Fractal Dimension-Based Filter — FDimBF —, based on this framework which uses the fractal dimension concept to perform redundancy analysis [7].

3.1 Fractals

Fractals are defined by the property of self-similarity, *i.e.* they present the same characteristics for different variations in scale and size. Thus, parts of the fractal, which may be a structure, an object or a data set, are similar, exact or statistically, to the fractal as a whole.

In general, fractals have unusual characteristics, such as the well known Sierpinsky Triangle in Figure 2(a) which has infinite perimeter and null area. Therefore, neither can be considered an Euclidean unidimensional object because it has infinite perimeter, nor a bidimensional Euclidean object as it presents a null area [8]. Consequently, a fractionary dimension may be considered, and is denoted Fractal Dimension. It should be observed that many real data sets behave like fractals. Hence, it is natural the idea of applying concepts from fractal theory to analyze these data sets [9].

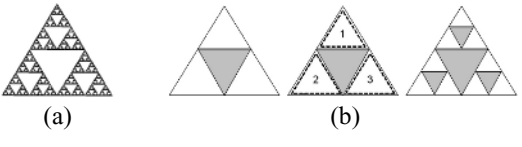


Fig. 2. (a) Triangle of Sierpinsky; (b) Constructing the triangle of Sierpinsky

3.2 Fractal Dimension of a Data Set

Fractal dimension can be associated to the idea of redundant features in a data set description, and the possibility of this data set being well described in a smaller dimension, *i.e.* using a subset of features. The main idea is to use the FD of the data set, which is relatively not affected by redundant features, as the criterion to determine how many and which are the most important features in the data set. In this way, the concepts of embedding dimension and intrinsic dimension should be defined. The first one is concerned with the number of features of the data set (its address space). However, the data set may be representing an object that has a smaller dimension than the one in which it is immersed. Thus, the intrinsic dimension is the spacial dimension of the object represented by the data set.

Conceptually, if a data set holds all its variables (features) independent one from the others, then its intrinsic dimension will be equal to the embedded dimension. However, whenever there is a correlation between two or more variables, the intrinsic dimension of the data set is reduced accordingly. Usually, neither correlations between features nor the existence of these correlations are known. By means of the intrinsic dimension of the data set it is possible to decide how many features are necessary to describe it. Different types of correlation may reduce the intrinsic dimension in different proportions, even by fractionary proportions. Hence, the concept of FD may be used as the intrinsic dimension of a data set [6].

There are several FD measures. Exactly self-similar fractals, *i.e.* the ones characterized by well defined construction rules, may have their FD calculated by $D = log(R)/log(\frac{1}{e})$, where R represents the quantity of parts and $\frac{1}{e}$ the scale in which the parts are generated at each iteration. For example, for the Triangle of Sierpinsky in Figure 2(a), $D = log(3)/log(2) = 1.58496$ since three parts in a $1{:}\frac{1}{2}$ scale are generated at each iteration, as shown in Figure 2(b).

Statistically self-similar fractals, such as real world data sets, may have their FD defined in several ways. One of them is the Correlation Fractal Dimension D_2 that can be calculated using the *Box-Count Plot* method [9]. This method consists in embedding the data set with a point set in an M-dimensional space, in an M-grid cells of side r. Afterwards, focusing on the i-th cell, the number of points that fall into each cell ($C_{r,i}$) is counted, and the value $S_2(r) = \sum_i {C_{r,i}}^2$ is computed. The Correlation Dimension D_2 is defined by $D_2 = \frac{\partial log(\sum_i S_2(r))}{\partial log(r)}$, $r \in [r_{min}, r_{max}]$. In theory, exactly self-similar fractals are infinite. In practice, real world data sets which present a finite number of points are considered statistically self-similar fractals for a determined interval of scales $r \in [r_{min}, r_{max}]$, if they fulfill a well known construction rule in this interval. Therefore, the intrinsic dimension of a specific data set may be measured by the slope of the linear part of the resulting graph obtained from plotting $S_2(r)$ for different values of r [6]. In this work, the correlation dimension D_2 will be simply denoted as fractal dimension D.

3.3 Algorithm FDimBF

As already mentioned, FDimBF selects features using a two step process:

Step 1: *relevance analysis* is performed to determine which features are important (relevant) w.r.t. the class labels, and the irrelevant features are removed.
Step 2: through *redundancy analysis*, redundant numeric features are determined and removed from the previous subset containing the relevant features, generating the final subset of selected features.

FDimBF is described by Algorithm 3.1, where relevance analysis can be accomplished by using any importance measure IM that enables to quantify how important a feature is w.r.t. the class labels.

Algorithm 3.1. *Fractal Dimension-Based Filter* — FDimBF

Require: $E = \{E_1, E_2, \ldots, E_N\}$, a data set with N examples described by M features $X = \{X_1, X_2, \ldots, X_M\}$ and labeled with their respective class labels $y_i, i = 1...N$, $y_i \in \{C_1, C_2, ...C_{N_{Cl}}\}$ of the class attribute Y
Ensure: $X_{opt} \subseteq X$, an "optimal" subset of relevant and non redundant features
1: *// Relevance analysis using the importance measure IM*
2: $X' = \emptyset$
3: **for all** $X_i \in X$ **do**
4: **if** X_i is relevant with respect to Y according to the importance measure IM **then**
5: $X' = X' \cup \{X_i\}$;
6: **end if**
7: **end for**
8: *// $X' \subseteq X$, such that X' contains the relevant features of data set E*
9: $L =$ set of M examples E described by the relevant features in X' according to the importance measure IM, *i.e.* without the class attribute Y;
10: *// Compute the fractal dimension D of data set L and find set X_{opt} of non redundant features*
11: $D = FractalDimension(L)$;
12: $X_{opt} = NonRedundantFeatures(L, X', D)$;
13: **Return** X_{opt}.

This step is carried out through lines 3–7. Note that this step evaluates features individually, thus having a low computational complexity. In the second step, lines 9–12, non redundant features are selected from the subset X' of the relevant features found in the first step. According to the FD, redundant features are defined as the ones that when excluded from the data set do not cause a significant modification in the recalculated FD value, called Partial Fractal Dimension pD. Search for non redundant features is carried out in a backward mode: first, the fractal dimension D is computed from the data set containing the

relevant features in X' — function *FractalDimension* in line 11 of Algorithm 3.1. After determining the FD of the subset of relevant features, it is necessary to establish which are the $\lceil D \rceil$ non redundant features. This is achieved by function *NonRedundantFeatures* in line 12 of Algorithm 3.1, described by Algorithm 3.2.

Algorithm 3.2. *NonRedundantFeatures*

Require: Data set L described by features in X' having FD=D
Ensure: $X_{opt} \subseteq X'$, subset of non redundant features according to the FD
1: $ListOfOrderedFeatures = [\ ]$;
2: **while** $X' \neq \emptyset$ **do**
3: $FeatDiff = \emptyset$;
4: **for all** $X_j \in X'$ **do**
5: Compute partial fractal dimension pD_{X_j} of feature X_j, *i.e.* considering all features in X' except feature X_j;
6: $FeatDiff = \{(X_j, D - pD_{X_j})\} \cup FeatDiff$;
7: **end for**
8: Select X_a from the set of elements $(X_i, D - pD_{X_i})$ of $FeatDiff$ such that $D - pD_{X_i}$ is the smallest;
9: *// Insert X_a at the head of the $ListOfOrderedFeatures$ list*
10: $ListOfOrderedFeatures = [X_a | ListOfOrderedFeatures]$;
11: $X' = X' - \{X_a\}$;
12: **end while**
13: X_{opt} is the set of $\lceil D \rceil$ first elements (features) from the $ListOfOrderedFeatures$ list;
14: **Return** X_{opt}.

Function *NonRedundantFeatures* calculates the pD value ignoring one attribute at a time. In other words, pD is computed considering all features in X' except the j-th feature under analysis. For each j-th feature not considered, the pD from the other features are computed aiming to find in each iteration the feature that shows the smallest difference between D and pD. This value indicates the contribution of this feature to characterize the data set. Thus, the feature that presents the minimum $D - pD$ value is inserted at the head of the ordered list *ListOfOrderedFeatures* and is removed from the set of relevant features. The time complexity of this FD calculation is $O(N.M^2)$, thus the total time complexity of FDimBF is $O(N.M^3)$. This process is repeated until there are no more features to analyze — lines 2–12 of Algorithm 3.2. Finally, the first $\lceil D \rceil$ elements (features) are selected from the list *ListOfOrderedFeatures*. These are the features having the greatest contribution to characterize the data set with fractal dimension D — line 13 of Algorithm 3.2.

We have implemented two different versions to carry out the first step of FDimBF: one performs relevance analysis using an information based measure, specifically the *information gain ratio* — FDimBF(1). The other one — FDimBF(2) — uses a distance based measure which ranks features using the

Manhattan distance. For both FDimBF(1) and FDimBF(2) the second step of analysis, *i.e.* redundancy analysis, is carried out by the Fractal Dimension Reduction — FDR — algorithm [6] which presents, as mentioned before, a time complexity of $O(N.M^2)$. The main idea of FDR is to discard features that have little influence over the fractal dimension of the data set, since the FD is relatively not affected by redundant features [6].

4 Empirical Study

Both versions of FDimBF were empirically evaluated and compared to four representative filter based feature selection algorithms in 11 data sets from UCI [10] — Table 1. Two of the chosen FS algorithms perform individual feature evaluation — ReliefF [11] and FCBF (*Fast Correlation-Based Filter*) [1] — and the other two select important features using subset evaluation — CFS (*Correlation-Based Feature Selection*) [12] and CBF (*Consistency-Based Filter* — CBF) [13]. ReliefF searches for nearest neighbors of examples with different class labels and features are weighed according to how well they differentiate these examples. This process is repeated m times and the time complexity of this algorithm is $O(m \cdot N \cdot M)$. FCBF carries on FS in two steps as FDimBF does, using an information measure in the second step to remove redundant features and presents a time complexity of $O(M^2)$. CFS evaluates the goodness of a subset of features by considering the individual predictive ability of each feature and the degree of correlation between them. Subset evaluation is also performed by CBF according to their inconsistency related to the class, searching for subsets that separate data to samples with a majority class. Both CFS and CBF present a time complexity of $O(N \cdot M^2)$. All algorithms were executed with default parameter values. Except for the FDimBF, they are available at Weka's environment [14].

For real world data, prior knowledge about important features is not often available, thus predictive accuracy of the constructed models is commonly used as an indirect measure to evaluate the quality of the selected features. In this work, for each of the 11 data sets, the original set of features and the subsets of features selected by each algorithm were evaluated by estimating through 10 fold cross-validation, the average error of the classifiers induced by $\mathcal{C}4.5$ [15], executed with its default parameters values.

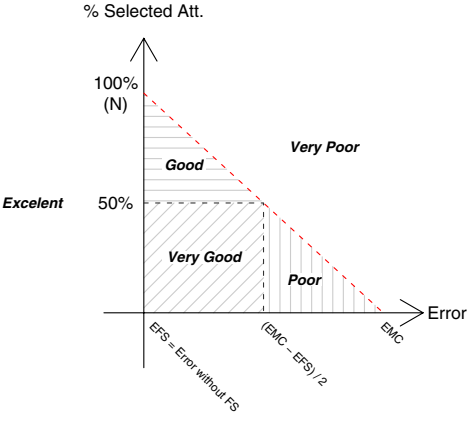


Fig. 3. Evaluation framework for FS algorithms

Experimental results, which are described in detail in [16], were analyzed under several frameworks [7].

Due to lack of space, we only present the experimental results under one of these frameworks which considers two main issues:(1) predictive accuracy of the model constructed using all features from the original data set, and the ones constructed using the subset of features selected by each FS algorithm, and (2) size of the subset of selected features in relation to the original data set.

In this framework, illustrated in Figure 3, we place the FS algorithms performance into five categories: excellent (△△△), very good (△△), good (△), poor (◇) and very poor (▽), where EFS stands for the model's error without FS and EMC represents the error of the data set's majority class whenever this error is less than 50%, otherwise EMC is set to 50%. The algorithms' performance according to this framework for the 11 data sets considered are presented in Table 1. This table also shows the number of examples (#Ex.), attributes (#Att.) and the EMC of each data set. Except for data sets Satimage, Segment, Vehicle and Waveform which have, respectively, 7, 7, 4 and 3 class labels, the others have only two classes.

Table 1. Data sets and performance of algorithms according to the percentage of selected attributes *versus* model's error

	#Ex.	#Att.	EMC	ReliefF	CFS	FCBF	CBF	FDimBF(1)	FDimBF(2)
Breast Cancer	683	9	34.48	—	—	—	△△△	△△△	△△△
Bupa	345	6	42.03	▽	◇	◇	◇	▽	△
German	1000	24	30.00	—	△△	△△△	△△△	△△△	△△△
Hungarian	261	10	36.05	—	△△△	△△△	△△△	△△	△△△
Ionosphere	34	351	35.90	△	△△	△	△△	△△	△△
Pima	769	8	34.98	—	△△	—	—	△△	▽
Satimage	4435	36	75.80	—	△△△	—	△△△	△△	△△
Segment	2310	19	85.70	△	△△	△	△△	△△	△△
Sonar	208	60	46.60	—	△△△	△△	△△	◇	△△
Vehicle	846	18	74.20	—	△	—	—	△△	△△
Waveform	5000	21	66.10	—	△△△	△△△	△	△△	△△
Excellent (△△△)				0	4	3	4	2	3
Very Good (△△)				0	4	1	3	7	6
Good (△)				2	1	2	1	0	1
Poor (◇)				0	1	1	1	1	0
Very Poor (▽)				1	0	0	0	1	1
All Selected Attributes (—)				8	1	4	2	0	0

As can be observed, both versions of FDimBF were the ones that obtained the total greatest number of excellent and very good performances, 9 out of 11. Algorithms CFS and CBF obtained respectively 8 and 7 excellent or very good performances, followed by 4 from FCBF. Poor performances occurred uniformly among all considered FS algorithms, and only 3 of them presented one very poor performance each: ReliefF and both versions of FDimBF. Regarding the number of important features selected, ReliefF selected all features in 8 of the 11 data sets. In fact, both versions of FDimBF were the only ones to always promote a reduction of the number of selected features for all data sets — last line in Table 1.

Therefore, under the proposed evaluation framework [7], from the 66 cases considered (11 data sets × 6 FS algorithms), 16 were excellent, 21 very good, 7 good, 4 poor, 3 very poor and 15 presented as subset of selected features the original ones. Thus, 66.67% of the cases were considered excellent, very good or good; 22.73% of the subsets selected were equal to the original set of features, and only 10.61% showed a poor or very poor performance. Hence, the majority of the FS algorithms contributed to improve both the reduction of the quantity of features and the accuracy of the constructed models under the proposed framework to evaluate the performance of FS algorithms.

5 Conclusion

In this work we proposed a feature selection algorithm that decouples relevance analysis and redundancy analysis. Two measures were considered for relevance analysis, one based on information and another based on distance. For redundancy analysis we proposed the use of a correlation measure based on the fractal dimension of the data set. Experimental results showed that the proposed algorithm is comparable to other FS algorithms. Furthermore, it showed capability to select small subsets of important features that enables one to construct models with a performance similar to well known algorithms such as CFS (*Correlation-Based Feature Selection*). Therefore, we consider that the FD is a good candidate to perform FS for supervised learning algorithms where to the best of our knowledge has not been used for this purpose.

Future work includes evaluation of the selected features from the domain expert's point of view, as well as the use of the area under the ROC curve instead of the error of the model constructed with the selected set of features.

References

1. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundance. Journal of Machine Learning Research **5** (2004) 1205–1224
2. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Massachusetts (1998)
3. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1-2) (1997) 273–324
4. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis **1**(3) (1997) 131–156 `http://dx.doi.org/10.1016/S1088-467X(97)00008-5`.
5. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In Kaufmann, M., ed.: Proc. of the 11th Int. Conf. on Machine Learning, CA (1994) 167–173
6. Traina, C., Sousa, E.P.M., Traina, A.J.M.: 24. In: Using Fractals in Data Mining. 1 edn. Volume 1 of 1. Wiley-IEEE Press, New Jersey, NJ (2005) 599–630
7. Lee, H.D.: Selection of important features for knowledge extraction from data bases. PhD thesis, Intitute of Mathematical Sciences and Computing — ICMC, University of São Paulo — USP, São Carlos (2005) (In Portuguese).

8. Sousa, E.P.M., Traina, C., Traina, A.J.M., Faloutsos, C.: How to use fractal dimension to find correlations between attributes. In: Workshop Notes of KDD 2002 Workshop on Fractals and Self-similarity in Data Mining: Issues and Approaches, Edmonton, Canada (2002) 26–30
9. Faloutsos, C., Kamel, I.: Beyond uniformity and independence: Analysis of r-trees using the concept of fractal dimension. In: Proc.of the 13th ACM SIGMOD Symposium on Principles of Database Systems, MN (1994) 4–13
10. Merz, C.J., Murphy, P.M.: UCI repository of machine learning datasets (1998)
11. Kira, K., Rendell, L.: A pratical approach to feature selection. In: Proc. of the 9th Int. Conf. on Machine Learning, Aberdeen, Scotland (1992) 249–256
12. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proc. of the 17th Int. Conf. on Machine Learning, CA, Morgan Kaufmann (2000) 359–366
13. Liu, H., Setiono, R.: A probabilistic approach to feature selection – a filter solution. In: Proc.of the 13th Int. Conf. on Machine Learning, Bari, Italy (1996) 319–327
14. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, CA (2000)
15. Quinlan, J.R.: $\mathcal{C}4.5$: Programs for Machine Learning. Morgan Kaufmann, CA (1988)
16. Lee, H.D., Monard, M.C., Voltolini, R.F., Wu, F.C.: Avaliação experimental e comparação de algoritmos de seleção de atributos importantes com o algoritmo FDimBF baseado na dimensão fractal. Technical Report 264, ICMC-USP (2005) `http://www.icmc.usp.br/~biblio/download/RT_264.pdf`.

Comparing Meta-learning Algorithms

Fabrício Enembreck and Bráulio Coelho Ávila

PUCPR, Pontifícia Universidade Católica do Paraná
PPGIA, Programa de Pós-Graduação em Informática Aplicada,
Rua Imaculada Conceição, 1155,
Curitiba PR - Brasil
{fabricio, avila}@ppgia.pucpr.br

Abstract. In this paper we compare the performance of KNOMA (Knowledge Mining Approach), a meta-learning approach for integration of rule-based classifiers, based on different rule inducers. Meta-learning approaches use a core learning algorithm for the generation of base classifiers that are further combined into a global one. This approach improves performance and scalability of data mining processes on large datasets. In a previous work we presented KNOMA, a meta-learning approach whose performance was evaluated using RIPPER as its core learning algorithm. Experiments have shown that the performance of KNOMA is comparable to that achieved with Bagging and Boosting. However, meta-learning is generally only sensitive to core algorithms used in the generation of base classifiers. KNOMA is a generic approach and can handle different rule-based inducers, although its advantages, drawbacks and use cases need to be precisely identified. We studied the variation of performance in the approach with base classifiers generated by two rule inducers (C45Rules and RIPPER) and also by C4.5. Interesting behaviors have been noticed in the experiments.

1 Introduction

Despite the augmentation of computer processing power, much attention is given to speeding up the data mining process. Basically, speeding up approaches can be either (i) data-oriented or (ii) algorithm-oriented. In (i) the dataset is processed and the learning instances space is reduced by discretization, attribute selection or sampling. In (ii) new search strategies are studied or data are mined in a distributed or parallel fashion. According to Freitas and Lavignton [10], distributed data mining consists of partitioning the data being mined among multiple processors, applying the same or different data mining algorithms to each local subset and then combining the local knowledge discovered by the algorithms into a global knowledge. The authors also discuss that such global knowledge is usually different (less accurate as the number of subsets increases) from the knowledge discovered by applying the mining algorithm on the entire dataset formed from the union of the individual local datasets. Unfortunately, subsets are often too large to be merged into a single dataset and then, some distributed learning approach must be used.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 289 – 298, 2006.

In a previous work we have presented a **model integration** technique based on a **meta-learning** approach. Based on rule sets learned separately from partitions of data, we build a single consistent rule set (this is model integration). This is accomplished by transforming each rule into a training instance. The predicting attributes are given by the tests of the antecedent of the rules and the classes by the consequents of the rules. Then, a learning algorithm generates a general model of rules (this is meta-learning). Because training instances represent pieces of knowledge, we titled such a technique KNOMA (**Kn**owledge **M**ining **A**pproach).

In [9] we have presented the results of KNOMA using the RIPPER rule induction as base learning algorithm. The results have shown that KNOMA is comparable with Bagging and Boosting and generates small sets of rules for several UCI datasets [3]. However a more detailed evaluation must be done in order to identify the power of the approach. Some of the questions that this paper will try to answer are: (i) can the approach be used jointly with different rule inducers? (ii) if "yes", what is the independency degree of the approach, that is, is the approach sensitive to the bias of each algorithm? The following section introduces some concepts on Distributed Data Mining and we review the main ideas of KNOMA in Section 3. We discuss our experiments and results in Section 4. Section 5 discusses some related works, advantages and drawbacks of the technique also pointing out possible improvements. Finally we conclude the paper with Section 6.

2 Distributed Data Mining

As discussed previously, very often, databases are too large to be mined. When this is the case, distributed data mining can be used for learning on knowledge bases or rule sets generated from parts of the entire training data set. Then, such rules sets can be used for classification of new instances in different ways. The quality of distributed data mining depends on many parameters such as the accuracy of the algorithms used for learning of the rule sets and, maybe the most important, the number of partitions in the training data. Chan and Stolfo [5] conclude in their paper that attacking the scaling problem of data reduction does have a negative impact on accuracy. That is, the larger the number of partitions, the lower the accuracy is. This probably happens because data sets distributed into different sites or sampled from the entire data set might let the learning algorithms discover inconsistent, incomplete and false concepts (rules). Thus, distributed data mining approaches generally must be able to deal with uncertain information. Common strategies for Distributed Data Mining as Vote [6], Multi-Scheme, Meta-learning [5] [13] [18], Bagging [4] [16] and Boosting [11] [15] have been discussed in [9].

3 The KNOMA Approach

We are looking for a method capable of generating understandable knowledge from distributed classifiers. Traditional methods combine only the output of base classifiers, producing several rule sets (knowledge sources) or knowledge about the classifiers behavior. Therefore, this knowledge cannot explain the data model and is not

useful for the knowledge discovery process. Furthermore, the technique must maintain or improve the accuracy of the model by producing a better classifier than any individual base classifier. We believe that a meta learner can use not only the output of the base classifiers as input, but it can use the concept descriptions of the base classifiers. Based on such an idea, we developed a method which (i) uses the knowledge of the base classifiers (generally represented in rules), (ii) generates a consistent single rule set and (iii) presents a good performance.

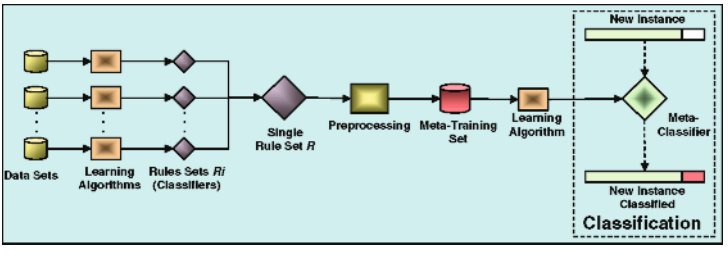


Fig. 1. The KNOMA process

The discovery of a consistent, understandable, accurate and small set of rules from a large rule set can be seen as a problem-solving technique. On the one hand, we can use many search strategies like heuristic search or genetic algorithms to find a "good" rule set. On the other hand, induction algorithms do exactly the same thing, that is, look for concepts in a space of candidate solutions. So, why not use the search strategy implemented inside an induction algorithm to discover a "good" rule set? Based on this idea we describe our technique in the next paragraphs.

The KNOMA approach (**Kno**wledge **M**ining **A**pproach), takes as input the N base classifiers learned from the distributed data sets or partitions of the training data. We assume that classifiers are classification rule sets R_i that contain the concepts and the classes definitions from the data. All rule sets are concatenated into a large single rule set R. Based on this rule set a preprocessing step is tackled for the generation of a meta-training set. Each meta-instance (instance of the meta-training set) represents a concept or rule. The attributes of this database are given by any existing test into the rules, that is, the antecedent part. The class of a meta-instance is given by the class predicted by the respective rule. Based on the meta-data, the learning algorithm is used for the generation of a meta-classifier that is used for the classification of new instances. Fig. 1 presents the entire process.

Our method is independent of any rule induction learning algorithm. With the process discussed in the former paragraphs we expect to generate a general model of the classifiers reasoning rather than a model of the classifiers actions. The advantage of this technique is that the meta-knowledge represented into the meta-classifier is indirectly a generalization of the raw training data and also explains the models and concepts of the distributed data sets. A meta-concept represents some patterns identified among the rules. When a test T is often used for the prediction of a class C into the rule set R, it will probably be used for the generation of a meta-concept which predicts the class C. The data transformations are sketched in the Fig. 2.

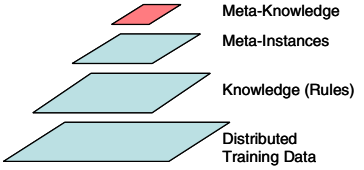


Fig. 2. The data transformations

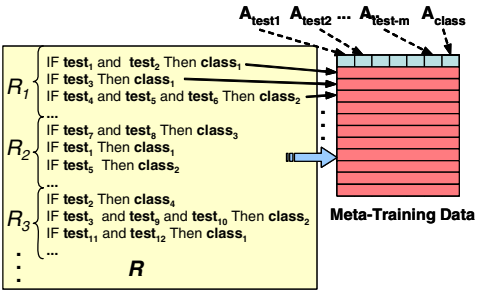


Fig. 3. Preprocessing

3.1 Preprocessing

A meta-instance is represented by a vector of Boolean variables, where each value tells about the presence or absence of a test. A test is a triple $\{A, O, V\}$, where A is the attribute, O is an operator member of $\{=, <, >, <=, >=\}$ and V is the attribute value.

We can observe in Fig. 3 that tests can appear repeatedly in different rules, since some concepts are not completely disjoint. In the preprocessing step, all tests are used as meta-attributes (attributes of the meta-training data set) and no treatment on the attribute values of the tests (V) is done. With this representation we try to generate a meta-model capable of describing the similarity between classifiers. Each meta-instance represents a generalization of the "raw" data (distributed data sets) and the meta-classifier learned from the meta-training set represents a generalization of meta-instances.

3.2 Filtering Meta-concepts Candidate

In the next paragraphs we will employ the following notation for a rule r_i: (r_i, *Tests* x *Class*), where r_i is the rule identifier, *Tests* is a set of tests like those of the former paragraph and *Class* is the class predicted by the rule. A *default rule*[1] r_i does not have tests and it is denoted by (r_i, *default* x *Class*).

Before learning the meta-classifier, we filter the meta-concepts candidate. A meta-concept candidate (*MCC*) is a rule that appears more than once and is not inconsistent. A rule (r_i, *Tests* x *C*) is not inconsistent when (r_j, *Tests* x *D*) $\notin R$, for all $j \neq i$, that is, there is not any rule into the rule set with the same antecedent predicting a different class. Although we let us fix the frequency a *MCC* to be higher than 1, this parameter (named *SF* – Selection Factor) can change according to the number of datasets. For instance, if we are mining a set of 10 datasets, $SF > 1$ and $SF <= 2$ or 3 seems adequate, while mining a set of 1000 datasets it probably is not so good. We think that a concept that appears in 20% of the rule sets should be reasonable.

This filtering procedure is especially useful for rule induction algorithms that generate "default rules" (rules that predict the default class). Since each rule set R_i

[1] A default rule predicts the default class.

presents a default rule, when all default rules of *R* predict the same class, with a great probability this concept will be generalized into a meta-concept by the induction algorithm. The previous identification of *MCCs* can aid the learning algorithm by reducing the search space, saving time and processing.

3.3 The KNOMA Algorithm

In this section we present a formal algorithm that implements the concepts discussed in the previous sections. Such an algorithm is presented in the Fig. 4.

KNOMA Algorithm

Input : *N* rule sets
Output: A small set of meta concepts ***MC-L***

1. Join the ***N*** rule sets into a single rule set ***R***
2. Move the MCCs from ***R*** to a list of MCCs, ***MCC-L***
3. Generate the meta-training set based on ***R***, run the rule induction algorithm and store the meta-concepts into a list, ***MC-L***
4. Append ***MCC-L*** to ***MC-L***
5. Prune rules from ***MC-L*** with low confidence or low support
6. Move default rules from ***MC-L*** to the end of the list
7. Return ***MC-L***

Fig. 4. The KNOMA algorithm

In the algorithm presented in Fig. 4, MCCs (Meta-Concepts Candidate) are moved to a list *MCC-L*. Such concepts are added to the end of the list of meta-concepts (*MC-L*) generated by the rule induction algorithm (line 4). Appending these concepts to *MC-L* we establish a priority between the rules, considering the most important the ones generated by the learn algorithm. This is necessary because we cannot guarantee that the rules of *MCC* and *MCC-L* are conceptually disjoint (cover disjoint sets of examples). *MC-L* rules are then pruned by removing some with *confidence* lower than 75% and *support* lower than 2 (line 5). For meta-concepts learned by the inducer, confidence and support are computed based on the number of examples covered by the rule and the number of examples correctly predicted by the rule in a standard way. For the meta-concepts candidate, the support is the frequency of the rule computed in the filtering step (MCCs always have 100% of confidence because they are consistent). The meta-concepts learned by the rule induction algorithm are disjoint between themselves and, therefore, there is no ordering between the concepts. Thus, default rules can appear anywhere inside the rule set. To make the interpretation of the rule set easier, we move the default rules to the end of the list by placing the one found by the learner before the one found as meta-concept candidate (whether it exists). Finally, MC-L can be used for the classification of new instances. In the next section we discuss some experiments done with the KNOMA algorithm.

4 Experiments

To test the KNOMA approach we have used 10 UCI databases [3] chosen arbitrarily and we compare results using two different rule inducers algorithms: RIPPER [8] and C45Rules [14]. The implementation of RIPPER is the one available in the Weka environment (JRIP) [17]. Both the algorithms have been tested with default values for parameters. We compare also the performance of KNOMA against Bagging and AdaBoosting [11] using 10 iterations for each one. All the results were 10-fold cross-validated. The KNOMA approach receives as input 10 base classifiers generated by the RIPPER for each iteration of the cross-validation using 90% of the training examples selected randomly. Then the performance is calculated on the test set. In such experiment we evaluated only the accuracy of the model. Tab. 1, Tab. 2 and Tab. 3 present the performance of the algorithms. Results in bold face represent the best performance. In Tab. 1 we can observe that KNOMA wins in 4 databases, while Bagging and Boosting win in 3 databases and the base classifier loose for any algorithm. KNOMA presents also small deviations.

Table 1. Comparing accuracy against RIPPER (JRIP)[2]

DB \ Alg.	KNOMA-RIP	RIPPER	Bagging-RIP	Boosting-RIP
Monk2	**66,67 +- 6,71**	58,58 +- 7,72	56,8 +- 9,11	62,13 +- 6,33
Audiology	79,72 +- 6,07	73,45 +- 11,30	77,87 +- 11,28	**82,74 +- 6,18**
Monk3	89,42 +- 9,65	86,06 +- 11,16	**90,98 +- 6,94**	89,34 +- 4,87
Ionosphere	**92,72 +- 5,05**	90,31 +- 5,14	92,02 +- 5,97	92,59 +- 4,43
Thyroid	**98,16 +-0,48**	97,08 +- 0,51	97,85 +- 0,42	98,00 +- 0,49
Tic-Tac-Toe	98,33 +- 1,33	97,07 +- 1,13	98,22 +- 1,32	**98,64 +- 1,69**
Iris	**98,00 +-3,05**	95,33 +- 6,70	96,00 +-4,42	94,67 +- 6,53
Soybean	86,37 +- 2,58	91,21 +- 3,71	92,02 +-2,41	**92,59 +- 2,78**
Monk1	68,59 +- 7,37	82,25 +- 13,75	**97,58 +- 4,92**	86,29 +- 11,24
Glass	68,92 +- 9,37	70,56 +- 8,49	**74,76 +-7,96**	74,30 +-10,50

For any database KNOMA outperforms bagging or boosting or both, never loosing for both of them. Maybe the most interesting results are the good performance on Monk2 and Iris and the very poor performance on Soybean, Monk1 and Glass. It is well-known that the Monk2 is a quite hard classification problem because it requires all the 6 predicting attributes be used to discriminate the classes. On the Iris database we observe a quite good performance and a very low deviation. The high error on the Glass database can be explained observing the base classifiers generated in the cross-validation. Tests like "(RI >= 1.5172)" and "(RI >= 1.51721)" appear in different rules and are considered as distinct attributes by KNOMA. However, they probably represent the same cut point. The effect of this is that many "meta attributes" are generated erroneously provoking an excessive partitioning of the data and breaking down relationships between the correct cut points. The Soybean database presents many values missing. Such behavior generates very different partitions for each iteration of the cross-validation process, generating very different classifiers too. In this case the problem of learning patterns between the classifiers is quite hard because

[2] Results extracted from [9].

there are no similarities. This also happens with the Monk1 database where classifiers generated with cross-validation do not have similarity apparent.

Table 2. Comparing accuracy against C45Rules and C45

DB \ Alg.	KNOMA-C45R	C45Rules	C45
Monk2	**62,13 +-2,77**	62,12 +-2,78	61,56 +-5,79
Audiology	67,31 +-7,13	51,51 +-13,37	**77,91 +-10,27**
Monk3	93,29 +-6,9	81,51 +-10.79	**93,47 +-7,23**
Ionosphere	**96,02 +-3,63**	93,84 +-2,24	92,83 +-3,13
Thyroid	97,03 +-0,75	96,41 +-1,89	**97,85 +-0,48**
Tic-Tac-Toe	73,16 +-2,40	70,42 +-7,22	**93,42 +-2,28**
Iris	**98,00 +-3,05**	95,99 +-4,42	96,29 +-4,56
Soybean	**96,19 +-2,38**	13,47 +-0,66	91,52 +-4,85
Monk1	68,72 +-15,18	**81,21 +-10,56**	71,30 +-11,44
Glass	47,62 +-4,47	66,2 +-5,83	**67,49 +-5,27**

We observe in Tab. 2 that KNOMA-C45R (KNOMA with C45Rules as base learning algorithm) clearly improves the performance of C45Rules, generating better results in 8 of 10 databases. Again, KNOMA does not perform well over Monk1 and Glass.

Table 3. Comparing accuracy against C45, Bagging and Boosting

DB \ Alg.	KNOMA-C45R	C45	Bagging-C45	Boosting-C45
Monk2	62,13 +-2,77	61,56 +-5,79	60,96 +-9,85	**64,48 +-5,30**
Audiology	67,31 +-7,13	**77,91 +-10,27**	81,97 +-8,22	**85,02 +-7,02**
Monk3	93,29 +-6,9	**93,47 +-7,23**	92,63 +-6,91	89,42 +-8,90
Ionosphere	**96,02 +-3,63**	92,83 +-3,13	93,18 +-4,00	93,17 +-3,38
Thyroid	97,03 +-0,75	97,85 +-0,48	**97,98 +-0,45**	97,56 +-0,53
Tic-Tac-Toe	73,16 +-2,40	93,42 +-2,28	92,17 +-2,92	**96,65 +-1,47**
Iris	**98,00 +-3,05**	96,29 +-4,56	95,33 +-5,20	93,33 +-6,67
Soybean	**96,19 +-2,38**	91,52 +-4,85	91,03 +-3,10	92,97 +-2,34
Monk1	68,72 +-15,18	71,30 +-11,44	73,53 +-10,64	**86,34 +-8,12**
Glass	47,62 +-4,47	67,49 +-5,27	72,79 +-10,40	**73,29 +-9,11**

Comparing results from Tab. 1 and Tab. 2 we observe that C45 outperforms RIPPER on most of the datasets used in the experiments. Bagging has a good effect in JRIP and not as good in C45 Tab. 3. As the results of C45 were better than those of C45Rules we compared the algorithms KNOMA-C45R and C45 in Tab.3. In Tab. 3 we clearly observe a performance improvement provoked by the Boosting technique in C45. As discussed in [11], Boosting is suitable when the core learning algorithm available is not stable, that is, the learning algorithm performs a local search in the tuple space and the classifiers generated change significantly even with small variations in training data. This occurs with C45. C45 does a local search selecting a single attribute at a time and then splitting the dataset into subsets. The process continues on each subset until a pure one (contains examples of a single class) is found. Thus, when C45 selects an inadequate attribute, the classifiers generated are generally more complex and less accurate regarding test data.

Moreover, Boosting seems to have a small effect in JRIP (Boosting-RIP). Such observations let us conjecture that RIPPER is more stable than C45 and more appropriate to noisy datasets. KNOMA-RIP has improved the performance of RIPPER on most databases but KNOMA-C45R did not present as good results. The stability of RIPPER and the instability of C45 explain the performances of KNOMA-RIP and KNOMA-C45R respectively. Cohen explains in [8] that RIPPER generates a rule starting from an empty conjunction. The rule is then "grown" by adding a new test, attempting to cover positive examples and rejecting negative examples. After a rule is grown, it is pruned. After the rules are generated, RIPPER improves the quality of the rule set by means of rule optimization. During optimization, any rule is evaluated/modified taking into account the performance of the entire rule set. Such global search strategy generally reduces the error generated by the step grow-and-simplify in the algorithm. Our observations coincide with Cohen's conjectures about the higher effectiveness of RIPPER on noisy datasets. Based on such ideas we conclude that the performance of KNOMA is highly sensitive to the stability of the core learning algorithm. We might approach this problem by pre-processing the input rules, taking into account the performance of the rule set over a validation data set, similar to the optimization rule procedure of RIPPER. Another approach is to weigh the rules. Input rules have generally different support and confidence levels but currently each rule represents a single meta-instance. A natural suggestion is to consider rules with high support and high confidence more important than less accurate or less significant ones. This could be achieved by, for instance, replicating such rules. However, such problem should be addressed in further refinements of KNOMA.

5 Related Work and Discussion

Many works have been previously developed to improve classification performance combining classifiers. In [1] and [2] the authors compare different ways of combining decisions of ensembles of classifiers. The authors focus on ensemble explanation, informing the user what rules are fired in the classification. However this partial explanation is not related to the concept description because rules can fire in different ensembles. Such process does not remove useless rules generating many candidate rules and confusing the understanding of the data model. Maybe the closest approach to ours are the works developed by Chan and Stolfo [5] [6] [7], which have studied the meta-learning approach, also discussed in [13] and [12]. The two first introduced two meta-learning techniques, the arbiter and the combiner. The former proposed a meta-learning approach for reducing the rules' complexity generated by inducers. In such a paper the meta-instances have the same structure (attributes) of the original data. Experiments showed the approach introduces some bias and does not work very well for any base-classifier. Very often researchers highlight the improvements in terms of performance attained with the combination of techniques such as bagging [4] and boosting [15]. However, such methods are applied to "well held" databases, which are not partitioned and that generally contain many data. In distributed data mining the data are partitioned into distributed, non-stratified, irregular and, probably, inconsistent databases. Furthermore, the distributed databases cannot be joined into a scalable database. We think that a distributed learning method can learn not only from

data, but also from the concept descriptions, reducing the scalability problem but also possibly reducing the classification accuracy. Despite this, the method introduced in this paper could improve the performance of base classifiers learned from distributed databases, producing understandable knowledge.

6 Conclusions

In this paper we addressed the knowledge integration problem, assessing the performance of KNOMA (Knowledge Mining Approach). KNOMA has been evaluated over different UCI databases by means of RIPPER and C45Rules' core learning algorithms and C45. KNOMA combines different base classifiers by producing meta-instances and running a learning algorithm against them. This process generates a meta-classifier which represents the similarities among the structures of base classifiers and consists in a small and understandable rule set. This method is very efficient because it takes concepts as input, rather than data. The technique exhibits a very good performance with RIPPER but not as good with C45Rules. Observing both the results obtained and related works from the literature, we can conjecture that such behavior is provoked by the instability of C45 decision trees used by C45Rules. We believe that KNOMA is sensitive to the stability of algorithms used in the generation of base classifiers. In the future, this issue should be approached with the use of rule selection techniques prior to the generation of the meta-instance database or by weighing rules using some quality measure.

References

1. Bernardini F. C. ; MONARD, M. C. . *Methods for Constructing Symbolic Ensembles from Symbolic Classifiers*. In: Congress of Logic Applied to Technology, 2005, Hyogo. Frontiers in AI and Applications. Netherlands : IOS Press, 2005. v. 132. p. 161-168.
2. Bernardini, F. C. ; MONARD, M. C. ; PRATI, R. C . *Constructing Ensembles of Symbolic Classifiers*. In: International Conference on Hybrid Intelligent Systems, 2005, Rio de Janeiro. Proceedings of the Fifth International Conference on Hybrid Intelligent Systems. California : IEEE Computer Society, 2005. v. 1. p. 315-320.
3. Blake. C., Merz. C. J., *UCI Repository of machine learning databases - www.ics. uci.edu/~mlearn/MLRepository.html*. Irvine. CA: University of California. Department of Information and Computer Science, 1998.
4. Breiman, L., *Bagging Predictors*, Machine Learning, n. 2, vol. 24, pp. 123-140, 1996.
5. Chan, P. K., Stolfo, S. J., *Experiments on multistrategy learning by meta-learning*, Proc. of the 2th International Conference on Information and knowledge management table of contents, pp. 314 – 323, Washington, D.C., United States, 1993. ISBN 0-89791-626-3
6. Chan, P. K., Stolfo, S. J., *A comparative evaluation of voting and meta-learning on partitioned data*, Proc. 12th Int. Conf. on Machine Learning, pp. 90-98, 1995.
7. Chan, P. K., Stolfo, S. J., Learning arbiter and combiner trees from partitioned data for scaling machine learning, Proc. First Int. Conf. Knowledge Discovery and Data Mining (KDD-95), pp. 39-44, 1995a.
8. Cohen, W. W., Fast effective rule induction. Proc. 20th. International Conference on Machine Learning, Tahoe City, CA, San Francisco: Morgan Kauffman, pp. 115-123, 1995.

9. Enembreck, F., Ávila, B. C., *KNOMA: A new Approach for Knowledge Integration*, 11th. IEEE Int. Symposium on Computers and Communications, Italy, 2006. to appear.
10. Freitas, A. A., Lavington, S. H., *Mining very large databases with Parallel Processing*, Kluwer Academic Publishers, The Netherlands, 1998. ISBN 0-7923-8048-7
11. Freund, Y., Shapire, R. E., *Experiments with a New Boosting Algorithm*, Proc. of International Conference on Machine Learning, pp. 148-156, 1996.
12. Kurgan, L., Reducing Complexity of Rule Based Models via Meta Mining, The 2004 Int. Conference on Machine Learning and Applications (ICMLA'04), 2004.
13. Prodromidis, A., Chan, P., Stolfo, S., *Meta-learning in distributed data mining systems: Issues and approaches*, In Advances in Distributed and Parallel Knowledge Discovery, H. Kargupta and P. Chan (editors), AAAI/MIT Press, 2000.
14. Quinlan. J. R. C4.5 : *Programs for Machine Learning*, Morgan Kauffman Publishers. San Mateo. California, 1993.
15. Shapire, R. E., *The Boosting Approach to Machine Learning: An Overview*, MSRI Worshop on Nonlinear Estimation and Classification, 2002.
16. Ting, K.M., Witten, I. H., *Stacking Bagged and Dagged Models*. ICML 1997: 367-375.
17. Witten, I. H., Frank, E., *Data Mining*, Morgan Kauffman Publishers, San Francisco, USA, 2000. ISBN 1-55860-552-5
18. Wolpert, D. H., *Stacked Generalization*, Neural Networks, n. 5, pp. 241-259, Pergamon Press, 1992.

A New Linear Dimensionality Reduction Technique Based on Chernoff Distance

Luis Rueda[1] and Myriam Herrera[2]

[1] Department of Computer Science and Center for Biotechnology
University of Concepción
Edmundo Larenas 215, Concepción, 4030000, Chile
lrueda@inf.udec.cl
[2] Department and Institute of Informatics
National University of San Juan
Cereceto y Meglioli, San Juan, 5400, Argentina
mherrera@iinfo.unsj.edu.ar

Abstract. A new linear dimensionality reduction (LDR) technique for pattern classification and machine learning is presented, which, though linear, aims at maximizing the Chernoff distance in the transformed space. The corresponding two-class criterion, which is maximized via a gradient-based algorithm, is presented and initialization procedures are also discussed. Empirical results of this and traditional LDR approaches combined with two well-known classifiers, linear and quadratic, on synthetic and real-life data show that the proposed criterion outperforms the traditional schemes.

1 Introduction

Pattern classification and machine learning constitute two important areas of artificial intelligence. These areas have been growing significantly in the past few years, and found many applications in medical diagnosis, bioinformatics, DNA microarray data analysis, security and many more. Designing a fast classifier, while maintaining a reasonable level of accuracy is a problem that has been well studied. Related to this is the well-known linear dimensionality reduction (LDR) problem, which has the feature of performing a simple linear algebraic operation, and being simpler to implement and understand. Various schemes that yield LDR have been reported in the literature for reducing to dimension one, including the well known *Fisher's discriminant analysis* (FDA) approach [5], *direct Fisher's discriminant analysis* [6], the *perceptron algorithm* (the basis of the back propagation *neural network* learning algorithms) [12], *piecewise recognition models* [11], *removal classification structures* [1], *adaptive linear dimensionality reduction* [9] (which outperforms Fisher's classifier for some data sets), *linear constrained distance-based classifier analysis* [4] (an improvement to Fisher's approach designed for hyperspectral image classification), *recursive Fisher's discriminant* [3], *pairwise linear classifiers* [15], and the *best hyperplane classifier* [13].

Consider the two-class classification problem with two classes, ω_1 and ω_2, represented by two normally distributed n-dimensional random vectors, $\mathbf{x}_1 \sim N(\mathbf{m}_1, \mathbf{S}_1)$ and $\mathbf{x}_2 \sim N(\mathbf{m}_2, \mathbf{S}_2)$, and whose *a priori* probabilities are p_1 and p_2 respectively. The

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 299–308, 2006.

aim is to linearly transform $\mathbf{x}_1$ and $\mathbf{x}_2$ into new normally distributed random vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ of dimension d, $d < n$, using a matrix $\mathbf{A}$ of order $d \times n$, in such a way that the classification error in the transformed space is as small as possible.

Let $\mathbf{S}_W = p_1\mathbf{S}_1 + p_2\mathbf{S}_2$ and $\mathbf{S}_E = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$ be the within-class and between-class scatter matrices respectively. The FDA criterion consists of maximizing the distance between the transformed distributions by finding $\mathbf{A}$ that maximizes the following function [5]:

$$J_F(\mathbf{A}) = tr\left\{(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}(\mathbf{A}\mathbf{S}_E\mathbf{A}^t)\right\} . \tag{1}$$

The matrix $\mathbf{A}$ that maximizes (1) is obtained by finding the eigenvalue decomposition of the matrix:

$$\mathbf{S}_F = \mathbf{S}_W^{-1}\mathbf{S}_E , \tag{2}$$

and taking the d eigenvectors whose eigenvalues are the largest ones. Since the eigenvalue decomposition of the matrix (2) leads to only one non-zero eigenvalue, $(\mathbf{m}_1 - \mathbf{m}_2)^t(\mathbf{m}_1 - \mathbf{m}_2)$, whose eigenvector is given by $(\mathbf{m}_1 - \mathbf{m}_2)$, we can only reduce to dimension $d = 1$.

Loog and Duin have recently proposed a new LDR technique for normally distributed classes [8], namely LD, which takes the Chernoff distance in the original space into consideration to minimize the error rate in the transformed space. They consider the concept of *directed distance matrices*, and a linear transformation in the original space, to finally generalize Fisher's criterion in the transformed space by substituting the within-class scatter matrix for the corresponding directed distance matrix. The LD criterion consists of minimizing the classification error in the transformed space by obtaining the matrix $\mathbf{A}$ that maximizes the function:

$$J_{LD_2}(\mathbf{A}) = tr\left\{(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} \left[\mathbf{A}\mathbf{S}_E\mathbf{A}^t - \mathbf{A}\mathbf{S}_W^{\frac{1}{2}}\frac{p_1\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}}) + p_2\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1p_2}\mathbf{S}_W^{\frac{1}{2}}\mathbf{A}^t\right]\right\} \tag{3}$$

The solution to this criterion is given by the matrix $\mathbf{A}$ that is composed of the d eigenvectors (whose eigenvalues are maximum) of the following matrix:

$$\mathbf{S}_{LD_2} = \mathbf{S}_W^{-1}\left[\mathbf{S}_E - \mathbf{S}_W^{\frac{1}{2}}\frac{p_1\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}}) + p_2\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1p_2}\mathbf{S}_W^{\frac{1}{2}}\right] . \tag{4}$$

The traditional classification problem has usually been solved by minimizing the error or, equivalently, by maximizing the separability between the underlying distributions using different criteria. The FDA criterion discussed above aims at minimizing the error by maximizing the Mahalanobis distance between distributions, resulting in an optimal criterion when the covariance matrices are coincident. In case the covariances are different, the optimal classifier is quadratic; the linear classification, in this case, results in maximizing the separability between the distributions by using a criterion that generalizes the Mahalanobis distance [7]. On the other hand, the LD criterion utilizes, as pointed out above, a directed distance matrix, which is incorporated in Fisher's criterion assuming the within-class scatter matrix is the identity.

In this paper, we take advantage of the properties of the Chernoff distance, and propose a new criterion for LDR that aims at maximizing the separability of the distributions in the transformed space. Since we are assuming the original distributions are normal, the distributions in the transformed space are also normal. Thus, the Bayes classifier in the transformed space is quadratic and deriving a closed-form expression for the classification error is not possible. Let $p(\mathbf{y}|\omega_i)$ be the class-conditional probability that a vector $\mathbf{y} = \mathbf{A}\mathbf{x}$ in the transformed space belongs to class ω_i. The Chernoff distance between two distributions, $p(\mathbf{y}|\omega_1)$ and $p(\mathbf{y}|\omega_2)$, is given as follows:

$$\text{Pr[error]} = \int p^{\beta}(\mathbf{y}|\omega_1)p^{1-\beta}(\mathbf{y}|\omega_2)d\mathbf{y} = e^{-k(\beta)}\,, \tag{5}$$

where

$$\begin{aligned} k(\beta) = \frac{\beta(1-\beta)}{2}(\mathbf{Am_1} - \mathbf{Am_2})^t[\beta\mathbf{AS}_1\mathbf{A} + (1-\beta)\mathbf{AS}_2\mathbf{A}]^{-1}(\mathbf{Am_1} - \mathbf{Am_2}) \\ + \frac{1}{2}\log\frac{|\beta\mathbf{AS}_1\mathbf{A} + (1-\beta)\mathbf{AS}_2\mathbf{A}|}{|\mathbf{AS}_1\mathbf{A}|^{\beta}|\mathbf{AS}_2\mathbf{A}|^{1-\beta}}\,. \end{aligned} \tag{6}$$

As it clearly follows from Equations (5) and (6), the smaller the error is, the larger the value of $k(\beta)$ is, and hence, in this paper, we propose to maximize (6), which indeed, provides a good approximation for the error.

2 Chernoff-Distance Linear Dimensionality Reduction

The criterion that we propose (referred to as the RH criterion) aims at maximizing the Chernoff distance between the transformed random vectors. Here, we consider $p_1 = \beta$ and $p_2 = 1-\beta$ (note that $\beta \in [0,1]$) since p_1 and p_2 "weight" the respective covariance matrices in the Chernoff distance. Since after the transformation, new random vectors of the form $\mathbf{y}_1 \sim N(\mathbf{Am}_1; \mathbf{AS}_1\mathbf{A}^t)$ and $\mathbf{y}_2 \sim N(\mathbf{Am}_2; \mathbf{AS}_2\mathbf{A}^t)$ are obtained, the aim is to find the matrix $\mathbf{A}$ that maximizes:

$$\begin{aligned} J^*_{c_{12}}(\mathbf{A}) = p_1p_2(\mathbf{Am}_1 - \mathbf{Am}_2)^t[\mathbf{AS}_W\mathbf{A}^t]^{-1}(\mathbf{Am}_1 - \mathbf{Am}_2) \\ + \log\left(\frac{|\mathbf{AS}_W\mathbf{A}^t|}{|\mathbf{AS}_1\mathbf{A}|^{p_1}|\mathbf{AS}_2\mathbf{A}|^{p_2}}\right)\,, \end{aligned} \tag{7}$$

which can also be written as follows (cf. [14]):

$$\begin{aligned} J^*_{c_{12}}(\mathbf{A}) = tr\left\{p_1p_2(\mathbf{AS}_W\mathbf{A}^t)^{-1}\mathbf{AS}_E\mathbf{A}^t \right. \\ \left. + \log(\mathbf{AS}_W\mathbf{A}^t) - p_1\log(\mathbf{AS}_1\mathbf{A}^t) - p_2\log(\mathbf{AS}_2\mathbf{A}^t)\right\} \end{aligned} \tag{8}$$

We now show[1] that for any value of $J^*_{c_{12}}(\mathbf{A})$, where the rows of $\mathbf{A}$ are linearly independent, there exists an orthogonal matrix $\mathbf{Q}$ such that the Chernoff distance in the transformed space is the same as that of the original space.

[1] The proof of all subsequent lemmas and theorems are given in the unabridged version of this paper [14].

Lemma 1. *Let* $\mathbf{A}$ *be any real* $d \times n$ *matrix,* $d \leq n$*, whose rows are linearly independent, and* $J^*_{c_{12}}(\mathbf{A})$ *be defined as in (8). Then*

$$max_{\{\mathbf{A}\}} J^*_{c_{12}}(\mathbf{A}) = max_{\{\mathbf{Q}:\mathbf{Q}\mathbf{Q}^t=\mathbf{I}_d\}} J^*_{c_{12}}(\mathbf{Q}) \tag{9}$$

In order to maximize $J^*_{c_{12}}$, we propose the following algorithm, which is based on the gradient method. The learning rate, one of the parameters to the algorithm are obtained by maximizing the objective function in the direction of the gradient. The first task is to find the gradient matrix using the operator ∇ as follows:

$$\begin{aligned}\nabla J^*_{c_{12}}(\mathbf{A}) = \\ & p_1 p_2 \left[\mathbf{S}_E \mathbf{A}^t (\mathbf{A}\mathbf{S}_W \mathbf{A}^t)^{-1} - \mathbf{S}_W \mathbf{A}^t (\mathbf{A}\mathbf{S}_W \mathbf{A}^t)^{-1} (\mathbf{A}\mathbf{S}_E \mathbf{A}^t)(\mathbf{A}\mathbf{S}_W \mathbf{A}^t)^{-1}\right]^t \\ & + \left[\mathbf{S}_W \mathbf{A}^t (\mathbf{A}\mathbf{S}_W \mathbf{A}^t)^{-1} - p_1 \mathbf{S}_1 \mathbf{A}^t (\mathbf{A}\mathbf{S}_1 \mathbf{A}^t)^{-1} - p_2 \mathbf{S}_2 \mathbf{A}^t (\mathbf{A}\mathbf{S}_2 \mathbf{A}^t)^{-1}\right]^t \end{aligned} \tag{10}$$

The procedure that maximizes $J^*_{c_{12}}$ is shown in Algorithm **Chernoff_LDA_Two**, which receives as a parameter, a threshold, θ, to indicate when the search will stop. Another parameter is the learning rate, η_k, which indicates how fast the algorithm will converge. To initialize $\mathbf{A}$ we use the result of FDA or LD that leads to the maximum value of the Chernoff distance in the transformed space. Also, as shown earlier in Lemma 1, once $\mathbf{A}$ is obtained, there always exists an orthogonal matrix $\mathbf{Q}$ such that $\mathbf{A}$ can be decomposed into $\mathbf{RQ}$ (cf. [14]). An additional step is then introduced in the algorithm, which decomposes $\mathbf{A}$ into $\mathbf{RQ}$, and utilizes $\mathbf{Q}$ in the next step.

Algorithm **Chernoff_LDA_Two**
Input: Threshold θ
begin
$\mathbf{A}^{(0)} \leftarrow \max_{\mathbf{A}}\{J^*_{c_{12}}(\mathbf{A}_F), J^*_{c_{12}}(\mathbf{A}_{LD})\}$ // Max. of FDA and LD
$k \leftarrow 0$
repeat
$\eta_k \leftarrow max_{\eta>0}\phi_{k_{12}}(\eta)$
$\mathbf{B} \leftarrow \mathbf{A}^{(k)} + \eta_k \nabla J^*_{c_{12}}(\mathbf{A}^{(k)})$
Decompose $\mathbf{B}$ into $\mathbf{R}$ and $\mathbf{Q}$
$\mathbf{A}^{(k+1)} \leftarrow \mathbf{Q}$
$k \leftarrow k + 1$
until $|J^*_{c_{12}}(\mathbf{A}^{(k-1)}) - J^*_{c_{12}}(\mathbf{A}^{(k)})| < \theta$
return $\mathbf{A}^{(k)}$, $J^*_{c_{12}}(\mathbf{A}^{(k)})$
end

Theorem 1. *Let* $\{\mathbf{A}^{(k)}\}_{k=1}^{\infty}$ *be the sequence of matrices generated by Algorithm* **Chernoff_LDA_Two**. *If* $\nabla J^*_{c_{12}}(\mathbf{A}^{(k)}) \neq 0$*, then* $J^*_{c_{12}}(\mathbf{A}^{(k)}) < J^*_{c_{12}}(\mathbf{A}^{(k+1)})$*. Otherwise, the algorithm terminates.*

Algorithm **Chernoff_LDA_Two** needs a learning rate, η_k, which depends on the case, and is related to the convergence/divergence problem. When η is small, convergence is slower but more likely. As opposed to this, when η is large, convergence is faster but the

algorithm could diverge. There are many ways of computing η_k, one of them being the expression that maximizes the value of $J^*_{c_{12}}$ in the next step [2]. Consider the following function of η:

$$\phi_{k_{12}}(\eta) = J^*_{c_{12}}(\mathbf{A}^{(k)} + \eta \nabla J^*_{c_{12}}(\mathbf{A}^{(k)}))\,. \tag{11}$$

Since the idea is to find $\mathbf{A}$ that maximizes $J^*_{c_{12}}(\mathbf{A})$, we choose the value of η that maximizes $\phi_{k_{12}}(\eta)$ as follows:

$$\eta_k = max_{\eta>0}\phi_{k_{12}}(\eta) = max_{\eta>0} J^*_{c_{12}}(\mathbf{A}^{(k)} + \eta \nabla J^*_{c_{12}}(\mathbf{A}^{(k)}))\,. \tag{12}$$

Although η_k is obtained using Equation (12), the former has to be computed using an iterative method, again. One of them is the secant method, as proposed in [2].

Starting from initial values of $\eta^{(0)}$ and $\eta^{(1)}$, we update η at step $j+1$ as follows:

$$\eta^{(j+1)} = \eta^{(j)} - \frac{\eta^{(j)} - \eta^{(j-1)}}{\frac{d\phi_{k_{12}}}{d\eta}(\eta^{(j)}) - \frac{d\phi_{k_{12}}}{d\eta}(\eta^{(j-1)})} \frac{d\phi_{k_{12}}}{d\eta}(\eta^{(j)})\,, \tag{13}$$

where $\frac{d\phi_{k_{12}}}{d\eta}$ is obtained by differentiating (11). This procedure is repeated until difference between $\eta^{(j-1)}$ and $\eta^{(j)}$ is as small as desired.

One of the important aspects when applying the secant algorithm is to find the initial values of η. We set one of them to $\eta_0 = 0$, while the other the value of η_1 is obtained from the angle difference between $\mathbf{A}$ at step k and the matrix obtained by adding the latter and the product between the learning rate and the gradient matrix, as per the following theorem.

Theorem 2. *Let $\phi_{k_{12}} : \mathbb{R}^{d\times n} \rightarrow \mathbb{R}$ be the continuously differentiable function defined in (11), where $J^*_{c_{12}}(\cdot)$ is defined in (8), and whose first derivative is given in (10). Then, the initial values of the secant method are given by $\eta_0 = 0$ and*

$$\eta_1 = \frac{d^2\epsilon - d}{tr\{\mathbf{A}^{(k)}[\nabla J^*_{c_{12}}(\mathbf{A}^{(k)})]^t\}}\,, \tag{14}$$

where $\epsilon = \cos\theta$, and θ is the angle between $\mathbf{A}^{(k)}$ and $[\mathbf{A}^{(k)} + \eta_k \nabla J^*_{c_{12}}(\mathbf{A}^{(k)})]$.

Theorem 2 can be geometrically interpreted in the following way. Since $\|\mathbf{A}\|_F$ is a norm that satisfies the properties of a metric, we can ensure that there exists a matrix norm $\|\mathbf{A}\|$ *induced* or *compatible* in $\mathbb{R}^n$, such that for any $\mathbf{A} \neq 0$, $\|\mathbf{A}\| = \sqrt{\lambda_1}$ holds, where λ_1 is the largest eigenvalue of $\mathbf{A}$ [2, pp.33]. Then, since that eigenvalue is $\lambda_1 = 1$, the matrix norm induced results in $\left\|\mathbf{A}^{(k)}\right\| = \left\|\mathbf{A}^{(k+1)}\right\| = 1$. In this way, we ensure that the rows of $\mathbf{A}^{(k)}$ and $\mathbf{A}^{(k+1)}$ reside in the same hypersphere in $\mathbb{R}^n$, whose radius is unity. Then, since those rows are linearly independent, they *could* be "rotated" independently using a vector[2], $\boldsymbol{\eta}$, of dimension d. However, Algorithm **Chernoff_LDA_Two** uses a scalar instead η. For this reason, the "rotation" can be seen on a hypersphere of radius d and all the rows of $\mathbf{A}$ are rotated using the same scalar. As an example, if we choose $\hat{\theta} = \pi/180$, and suppose that $\mathbf{A}^{(k)}$ is of order $1 \times n$, i.e.

[2] Using a vector, $\boldsymbol{\eta}$, to update $\mathbf{A}^{(k)}$ is beyond the scope of this paper, and is a problem that we are currently investigating.

a vector in $\mathbb{R}^n$, we obtain a value of $\epsilon \approx 0.9998$. Thus the variation between $\mathbf{A}^{(k)}$ and $\mathbf{A}^{(k+1)}$ is one degree, where, obviously, the value of η_1 depends also on $\mathbf{A}^{(k)}$ and $\nabla J^*_{c_{12}}(\mathbf{A}^{(k)})$.

3 Simulations on Synthetic Data

The tests on synthetic data involve ten datasets of dimensions $n = 10, 20, \ldots, 100$ each with two randomly generated normally distributed classes. The two classes of each dataset, ω_1 and ω_2, are then fully specified by their parameters, $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{S}_1$ and $\mathbf{S}_2$. We also randomly generated $p_1 = [0.3, 0.7]$ and assigned $p_2 = 1 - p_1$. We trained three LDR techniques, FDA, LD and RH (the method proposed in this paper), using these parameters, and for each dataset we generated 100,000 samples for testing purposes. For each dataset, we found the corresponding transformation matrix $\mathbf{A}$ for each dimension $d = 1, \ldots, n-1$. After the linear transformation is performed, we tested two classifiers: the linear classifier, which is obtained by averaging the covariances matrices in the transformed space, and the quadratic classifier which is the one that minimizes the error rate assuming that the parameters in the transformed data are given by $\mathbf{A}\mathbf{m}_i$ and $\mathbf{A}\mathbf{S}_i\mathbf{A}^t$.

Table 1. Error Rates for the three classifiers, FDA, LD and RH, where the samples are projected onto the d^*-dimensional space with d^* gives the lowest error rate for $d = 1, \ldots, n-1$

	FDA+Q		LD+Q		RH+Q		FDA+L		LD+L		RH+L	
n	error	d^*	error	d^*	error	d^*	error	d^*	error	d^*	error	d^*
10	0.286530	1	0.053140*	9	0.053230	9	0.289790	1	0.288820*	6	0.288830	9
20	0.222550	1	0.019680	18	0.019580*	18	0.227000	1	0.220180	3	0.218780*	4
30	0.151190	1	0.002690*	24	0.002690*	24	0.182180*	1	0.182480	27	0.182480	27
40	0.287250	1	0.006600	36	0.006570*	36	0.297840	1	0.295370	8	0.294660*	6
50	0.370450	1	0.005490*	49	0.005490*	49	0.396160*	1	0.397450	1	0.397450	1
60	0.320760	1	0.000680*	56	0.000680*	56	0.322920	1	0.316030	21	0.315250*	23
70	0.381870	1	0.000010*	28	0.000010*	28	0.381960	1	0.381910*	30	0.381910*	30
80	0.323140	1	0.000000*	37	0.000000*	37	0.342980	1	0.334170	23	0.334080*	25
90	0.324740	1	0.000000*	30	0.000000*	30	0.326360	1	0.324740*	1	0.324740*	1
100	0.198610	1	0.000000*	31	0.000000*	31	0.278590*	1	0.278730	78	0.278720	72

The minimum error rates obtained for each individual classifier for synthetic data are shown in Table 1. The first column represents the dimension of each datset. The next columns correspond to the error rate and the *best* dimension d^* for the three LDR methods and for each classifier, quadratic and linear. The '*' symbol beside the error rate indicates that the lowest among the three methods, FDA, LD and RH, was obtained. Note that for FDA, $d^* = 1$, since, as pointed out earlier, the objective matrix contains only one non-zero eigenvalue. We observe that for the quadratic classifier LD and RH outperformed FDA for all the datasets. Also, LD and RH jointly achieved minimum error rate for seven datasets, while RH obtained the best error rate in nine out of ten

Table 2. Error rates for the quadratic and linear classifiers in the one-dimensional space, where the transformed data was obtained using the FDA, LD and RH methods

n	FDA+Q	LD+Q	RH+Q	FDA+L	LD+L	RH+L
10	0.286530	0.169750	0.154790*	0.289790*	0.320460	0.385010
20	0.222550	0.218260	0.204680*	0.227000	0.229260	0.222490*
30	0.151190	0.022950*	0.022950*	0.182180*	0.277120	0.277120
40	0.287250	0.219680	0.219590*	0.297840*	0.458030	0.458030
50	0.370450	0.237150	0.237080*	0.396160*	0.397450	0.397450
60	0.320760	0.122350*	0.122440	0.322920*	0.440710	0.440710
70	0.381870	0.061530*	0.061530*	0.381960*	0.402320	0.402320
80	0.323140	0.060320*	0.060320*	0.342980*	0.444530	0.444530
90	0.324740	0.087150*	0.087150*	0.326360	0.324740*	0.324740*
100	0.198610	0.093410*	0.093410*	0.278590*	0.332370	0.332370

datasets. For the linear classifier, again, LD and RH outperformed FDA, and also RH achieved the lowest error rate in six out of ten datasets, outperforming LD.

In Table 2, the results for the dimensionality reduction and classification for dimension $d = 1$ are shown. For the quadratic classifier, we observe that as in the previous case, LD and RH outperformed FDA, and that the latter did not obtain the lowest error rate in any of the datasets. On the other hand, RH yielded the lowest error rate in nine out of ten datasets, outperforming LD. FDA, however, did perform very well for the linear classifier, achieving the lowest error rate in eight out of ten datasets. RH, though not the best, outperformed LD yielding the lowest error rate in two out of ten datasets. Note also that the good performance of FDA and the linear classifier is due to the fact that the optimal Bayes classifier for normal distributions is linear when the covariances are coincident.

4 Experiments on Real-Life Data

We also performed a few simulations on real life data which involve 44 two-class, datasets from the UCI repository [10]. Six datasets were of two classes, and from

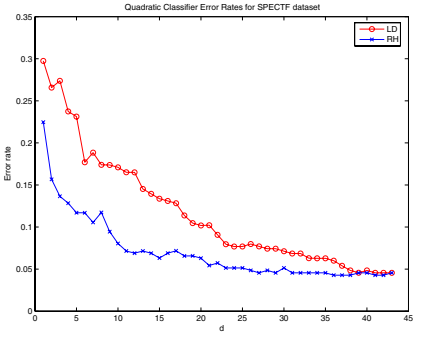


Fig. 1. Quadratic classifier error rates, SPECTF

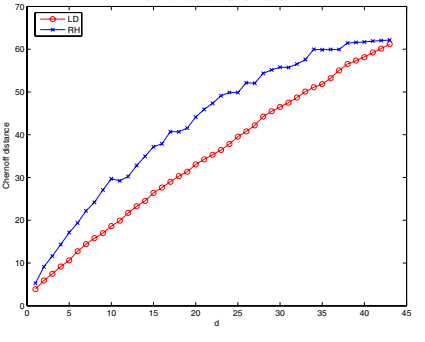


Fig. 2. Chernoff distance for SPECTF

Table 3. Error rates for the two-class datasets drawn from the UCI machine learning repository

	FDA+Q		LD+Q		RH+Q		FDA+L		LD+L		RH+L	
Dataset	error	d^*	error	d^*	error	d^*	error	d^*	error	d^*	error	d^*
W	0.030754	1	0.027835*	1	0.030754	1	0.039621	1	0.038150*	6	0.039621	1
B	0.362017	1	0.388571	4	0.353613*	1	0.309916	1	0.330168	5	0.301261*	5
P	0.226435*	1	0.251265	2	0.226435*	1	0.229033*	1	0.230383	7	0.229033*	1
D	0.031522*	1	0.040266	27	0.031522*	1	0.042079	1	0.029889*	20	0.036785	28
C	0.164943	1	0.168276	11	0.161379*	11	0.161609	1	0.158391	8	0.144828*	5
S	0.247773	1	0.045588	41	0.042810*	36	0.233646	1	0.176373*	19	0.180378	15
I,1,2	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1
I,1,3	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1
I,2,3	0.050000	1	0.030000*	1	0.040000	2	0.030000*	1	0.040000	1	0.030000*	1
T,1,2	0.021637	1	0.010819	4	0.005263*	3	0.059357	1	0.032749	4	0.027193*	4
T,1,3	0.022222*	1	0.027778	2	0.027778	2	0.038889	1	0.027778*	4	0.027778*	4
T,2,3	0.000000*	1	0.000000*	2	0.000000*	1	0.000000*	1	0.000000*	4	0.000000*	1
G,1,2	0.310000*	1	0.397619	7	0.397619	8	0.281905*	1	0.295714	8	0.289048	7
G,1,3	0.223611	1	0.204167	1	0.112500*	8	0.223611	1	0.204167	1	0.161111*	8
G,1,5	0.000000*	1	0.000000*	5	0.000000*	1	0.000000*	1	0.000000*	5	0.000000*	1
G,1,7	0.020000*	1	0.040000	8	0.020000*	1	0.040000	1	0.030000*	1	0.040000	1
G,2,3	0.158611	1	0.213333	8	0.153611*	8	0.158611*	1	0.167222	4	0.166111	8
G,2,5	0.109722	1	0.098333*	7	0.098333*	6	0.099722	1	0.088333*	7	0.088333*	6
G,2,7	0.027273*	1	0.063636	7	0.027273*	1	0.046364	1	0.037273	8	0.018182*	8
G,3,5	0.000000*	1	0.000000*	1	0.000000*	1	0.025000	1	0.000000*	6	0.000000*	7
G,3,7	0.060000	1	0.020000*	2	0.040000	4	0.060000*	1	0.060000*	1	0.060000*	1
G,5,7	0.050000*	1	0.070000	4	0.050000*	1	0.050000	1	0.050000	8	0.025000*	2
N,1,2	0.007143	1	0.007692	6	0.000000*	6	0.007692	1	0.007143*	11	0.007692	1
N,1,3	0.000000*	1	0.000000*	3	0.000000*	1	0.000000*	1	0.000000*	3	0.000000*	1
N,2,3	0.016667	1	0.016667	3	0.008333*	7	0.016667	1	0.008333*	12	0.016667	1
J,1,2	0.001435*	1	0.005263	3	0.001435*	1	0.001435*	1	0.001435*	11	0.001435*	1
J,1,3	0.000370*	1	0.001108	7	0.000370*	1	0.001108*	1	0.001108*	11	0.001108*	1
J,4,5	0.007512	1	0.001778*	7	0.004865	3	0.004417	1	0.000881*	9	0.004861	1
J,6,7	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1	0.000000*	1
J,8,9	0.066800	1	0.051309*	11	0.052896	6	0.069473	1	0.071601	11	0.068404*	8
L,C,G	0.083547	1	0.051096	15	0.047083*	10	0.083547	1	0.084903	12	0.081574*	6
L,D,O	0.033400	1	0.015402	15	0.014777*	10	0.032784	1	0.030216*	14	0.032776	12
L,J,T	0.009741	1	0.004520	10	0.003875*	8	0.009741	1	0.009741	15	0.009087*	10
L,K,R	0.098878	1	0.041405*	12	0.042081	10	0.096207	1	0.095522	13	0.094207*	1
L,M,N	0.031751	1	0.015847	13	0.014590*	13	0.034936	1	0.033033*	13	0.034936	1
L,O,Q	0.045591*	1	0.057280	11	0.046253	1	0.046237	1	0.050133	11	0.045583*	9
L,P,R	0.020505	1	0.012176	9	0.010248*	9	0.022432	1	0.021787*	7	0.022428	6
L,U,V	0.010748	1	0.007595	15	0.006966*	9	0.012018	1	0.011381*	10	0.011381*	9
L,V,W	0.027057	1	0.027048	15	0.022438*	10	0.029706	1	0.031035	13	0.028381*	5
E,1,2	0.003051	1	0.001312	10	0.000873*	10	0.006556*	1	0.006556*	10	0.006556*	1
E,3,4	0.002277	1	0.002277	1	0.002273*	8	0.002277*	1	0.002277*	1	0.002277*	1
E,5,6	0.001370	1	0.000457	6	0.000000*	8	0.001826	1	0.002283	11	0.001822*	13
E,7,8	0.000911	1	0.000455*	3	0.000455*	3	0.000911	1	0.000455*	1	0.000911	1
E,9,10	0.011357	1	0.000472*	12	0.000943	12	0.012300	1	0.009933	11	0.008518*	6

multi-class datasets, we extracted pairs of classes. The priors were estimated as $p_i = n_i/(n_i + n_j)$, where n_i and n_j are the number of samples for classes ω_i and ω_j respectively. Three LDR techniques, FDA, LD, and RH, were trained and the mean of the error rate computed for quadratic (Q) and linear (L) classifiers in a ten-fold cross-validation manner. The errors for the best value of d, $d = 1, \ldots, n$, are shown in Table 3. The first column indicates the name of the dataset and the classes separated by "," (when classes are not given, it means the problem itself is two-class), where the name of the dataset is as follows: W = Wisconsin breast cancer, B = Bupa liver, P = Pima, D = Wisconsin diagnostic breast cancer, C = Cleveland heart-disease, S = SPECTF heart, I = Iris, T = Thyroid, G = Glass, N = Wine, J = Japanese vowels, L = Letter, and E = Pendigits. For the quadratic classifier, RH outperformed both FDA and LD, since the former obtained the lowest error rate in 34 out of 44 cases, while FDA and LD obtained the lowest error rate both in 17 and 16 cases respectively. In the case of the linear classifier, RH also outperformed FDA and LD – the former was the best in 31 cases, while the latter two in 15 and 26 cases respectively. In this case, although RH is coupled with a linear classifier, while it optimizes the Chernoff distance and is expected to work well with a quadratic classifier, RH obtained the lowest error rate in more cases than LD. Also, for the quadratic classifier, on datasets B, P, S and G,1,3, the error rate yielded by RH is significantly smaller than that of LD. In particular, on the G,1,3 dataset, the difference between RH and LD is approximately 9% and with respect to FDA is more than 10%. For the linear classifier on the same dataset, RH is also more accurate than FDA and LD with a difference of approximately 4% and 6% respectively.

We also plotted the error rate and Chernoff distance for the SPECTF dataset for all values of d, for LD and RH. FDA was excluded, since as pointed out earlier, the data can only be transformed to dimension 1. The corresponding plots for the error of the quadratic classifier and the Chernoff distances are depicted in Figs. 1 and 2 respectively. The error rate (in general) decreases as the dimension d of the new space increases. Also, in this case, the RH clearly leads to a lower error rate than LD, while both converge to similar error rates for values of d close to n. This reflects the fact that as the Chernoff distance in the transformed space increases (see Fig. 2), the error rate of the quadratic classifier decreases.

5 Conclusion

We have presented a new criterion for linear dimensionality reduction (LDR), which aims at maximizing the Chernoff distance in the transformed space. The criterion is maximized by using a gradient-based algorithm, for which the convergence and initialization proofs are discussed. The proposed method has been shown to outperform traditional LDR techniques, such as FDA and LD, when coupled with quadratic and linear classifiers on synthetic and real-life datasets. The method is powerful enough to be intuitively extended for the multi-class scenario, which we are currently formalizing and testing.

References

1. M. Aladjem. Linear Discriminant Analysis for Two Classes Via Removal of Classification Structure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(2):187–192, 1997.
2. E. Chong and S. Zak. *An Introduction to Optimization*. John Wiley and Sons, Inc., New York, NY, 2nd edition, 2001.
3. T. Cooke. Two Variations on Fisher's Linear Discriminant for Pattern Recognition. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 24(2):268–273, 2002.
4. Q. Du and C. Chang. A Linear Constrained Distance-based Discriminant Analysis for Hyperspectral Image Classification. *Pattern Recognition*, 34(2):361–373, 2001.
5. R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, NY, 2nd edition, 2000.
6. H. Gao and J. Davis. Why Direct LDA is not Equivalent to LDA. *Pattern Recognition*, 39:1002–1006, 2006.
7. M. Herrera and R. Leiva. Generalización de la Distancia de Mahalanobis para el Análisis Discriminante Lineal en Poblaciones con Matrices de Covarianza Desiguales. *Revista de la Sociedad Argentina de Estadística*, 3(1-2):64–86, 1999.
8. M. Loog and P.W. Duin. Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):732–739, 2004.
9. R. Lotlikar and R. Kothari. Adaptive Linear Dimensionality Reduction for Classification. *Pattern Recognition*, 33(2):185–194, 2000.
10. D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998. University of California, Irvine, Dept. of Computer Science.
11. A. Rao, D. Miller, K. Rose, , and A. Gersho. A Deterministic Annealing Approach for Parsimonious Design of Piecewise Regression Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):159–173, 1999.
12. S. Raudys. Evolution and Generalization of a Single Neurone: I. Single-layer Perception as Seven Statistical Classifiers. *Neural Networks*, 11(2):283–296, 1998.
13. L. Rueda. Selecting the Best Hyperplane in the Framework of Optimal Pairwise Linear Classifiers. *Pattern Recognition Letters*, 25(2):49–62, 2004.
14. L. Rueda and M. Herrera. Linear Discriminant Analysis by Maximizing the Chernoff Distance in the Transformed Space. *Submitted for Publication*, 2006. Electronically available at http://www.inf.udec.cl/∼lrueda/papers/ChernoffLDAJnl.pdf.
15. L. Rueda and B. J. Oommen. On Optimal Pairwise Linear Classifiers for Normal Distributions: The Two-Dimensional Case. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 24(2):274–280, February 2002.

A Machine Learning Approach to the Identification of Appositives

Maria Claudia Freitas[1], Julio C. Duarte[2], Cícero N. Santos[3], Ruy L. Milidiú[3], Raúl P. Rentería[3], and Violeta Quental[1]

[1] Departamento de Letras, Pontifícia Universidade Católica, Rio de Janeiro, Brazil
claudiaf@let.puc-rio.br, violetaq@rdc.puc-rio.br
[2] Centro Tecnológico do Exército, Rio de Janeiro, Brazil
jduarte@ctex.eb.br
[3] Departamento de Informática, Pontifícia Universidade Católica, Rio de Janeiro, Brazil
milidiu@inf.puc-rio.br, nogueira@inf.puc-rio.br, renteria@inf.puc-rio.br

Abstract. Appositives are structures composed by semantically related noun phrases. In Natural Language Processing, the identification of appositives contributes to the building of semantic lexicons, noun phrase coreference resolution and information extraction from texts. In this paper, we present an appositive identifier for the Portuguese language. We describe experimental results obtained by applying two machine learning techniques: Transformation-based learning (TBL) and Hidden Markov Models (HMM). The results obtained with these two techniques are compared with that of a full syntactic parser, PALAVRAS. The TBL-based system outperformed the other methods. This suggests that a machine learning approach can be beneficial for appositive identification, and also that TBL performs well for this language task.

1 Introduction

Appositives are structures composed by semantically related noun phrases (NPs)[1]. For linguistic units to be in apposition, they must be identical in reference, that is, they must be co-referential. In Natural Language Processing (NLP), the identification of appositives contributes to the elaboration of semantic lexicons [1,2], noun phrase co-reference resolution [3,4,5] and information extraction, since, semantically, pairs of involved NPs express an explicative relationship.

Appositive structures are broad enough to be applied to different domains and text genres. Hence, they are also of interest in Information Extraction (IE), as they accomplish the portability requirement of IE systems. Semantically, an apposition frequently expresses equality, as illustrated in (1), or a set membership relation as illustrated in (2).

[1] Appositives can also take the form of PPs (when the head of the NP is omitted. See Section 2) or be expressed by clauses with a nominal value. We are not treating these cases here.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 309–318, 2006.

1 João Gilberto, the father of Bossa Nova, played yesterday....
2 A woman, Angela Merkel, was elected ....
3 The Finance Minister, Antonio Palocci, argued that ...

However, sometimes the difference between equality and set membership is blurred. In (3), *Antonio Palocci* and *Finance Minister* express an identity relation, but *Antonio Palocci* is a *Finance Minister*, i.e., the two NPs convey also a membership relation.

Although the explicative apposition is always between commas, this feature is not sufficient to deterministically identify it, since adverbial structures (4) and coordination (5) also appear between commas:

4 ... decided by the TJSP, the complex nature of the process, justified by the...
5 ... were the philosopher José Arthur Giannotti, the historian Boris Fausto, ...

Traditionally, due to its complex structure, appositive identification is realized through full syntactic parsing, a costly solution if we consider the amount of texts to be parsed for an IE task, for instance. Therefore, the development of shallow parsing methods to identify apposition seems a profitable way to deal with the problem.

As far as we know, besides the parser PALAVRAS [6], the work presented here is the only one related to Portuguese appositive identification. Still, when using PALAVRAS, this identification requires a full syntactic parsing. Nevertheless, apposition does not appear as a relationship between two NPs, but as a chunk. That is, a given NP is tagged as an appositive, but no other NP is identified as related to it.

For the English language, appositive identification is one of the basic tasks in complex NLP systems. In [3] and [5] appositive identification is included as a feature to help the co-reference resolution task. As noted by the authors, they use a simple and overtly restricted heuristic: there is an apposition when the NP is surrounded by commas, contains an article, and is immediately preceded by another NP. As a result, structurs like in (4) and (5) would be wrongly identified. A very similar heuristic is used in [1], but their general goal is the automatic generation of a semantic lexicon. As a consequence of this indirect treatment of appositive structures, it is not possible to directly compare precision and recall.

We present here the results of five experiments designed to address appositive identification: in the first experiment, a set of handcrafted rules is used. In the second, a plain HMM model is tested. In the third, a HMM followed by handcrafted rules is employed. In the fourth and fifth experiments we use TBL, having as a initial classifier a plain HMM and a set of handcrafted rules, respectively.

The results of each experiment were evaluated and compared with the results of a full syntactic parser, PALAVRAS. The TBL-based extractor outperformed the other methods. This suggests that a Machine Learning (ML) approach can be beneficial for appositive identification, and also that TBL performs very well for this kind of language task.

The remainder of the paper is organized as follows. In the following section, the grammatical model of the appositive structure is presented. In section 3, the techniques used in the experiments, that is, HMM and TBL are described. In section 4, the experimental design is described. In section 5, an evaluation of the experiments is presented, comparing the results with the ones obtained through full parsing analysis. Finally, in section 6, we present our concluding remarks.

2 Appositives

In general terms, the appositive is defined as a nominal term that joins another for the purposes of explanation. Although grammarians may vary in how they use the term *apposition*, we limit ourselves here to what [7] call *full, strict and restrictive apposition*. This means that we are looking for structures in which:

- each of the appositives can be separately omitted without affecting the resulting sentence (full)
- the appositives belong to the same syntactic class (both are NPs) (strict)
- the appositives are in separate information units (nonrestrictive).

Besides, for our purposes, appositives must contain at least one Proper Noun, and we discarded the apposition which begins with the preposition *of*, a kind of implicit apposition in which the head of the noun phrase is omitted:

6 Paul Wolfowitz, of Banco Mundial, was indicated... (employee)
7 President Lula, of Brazil, made a speech... (origin)
8 The book Vidas Secas, of Graciliano Ramos, is considered... (authorship)

Although very common in newspaper texts, the kind of relation expressed in these constructions can only be understood with world's knowledge. In fact, it is as if the preposition *of* had a wildcard value to be used according to the circumstances. In the preceding examples, in each occurrence of *of* there is a distinct semantic relation. The relations extracted in cases (6), (7) and (8) would not be significant taken out of context:

6' President Lula => Brazil
7' Paul Wolfowitz => Banco Mundial
8' The book Vidas Secas => Graciliano Ramos

It is worth noting that apposition is a linguistic phenomenon in which two structures participate in a relationship, which means that it may not have a head. According to [7], in nonrestrictive apposition, the two appositive units contribute to express relatively independent information, with the first appositive acting as the defined expression and the second as a definer. So, in a sentence like:

10 Robin, Batman's sidekick, was Dick Grayson.

Robin is the defined expression and *Batman's sidekick* is the definer. And, in a sentence like

11 Batman's sidekick, Robin, was Dick Grayson.

we have an inversion: *Batman's sidekick* is the definer, and *Robin* the defining expression. This led us to a tricky situation: although the difference between definer and defined expression may be informative on the discourse level, regarding new and old information in the utterance, it seems odd, if we are planning to build a semantic hierarchy, for example, to extract data such as:

- Batman's sidekick => Robin
- Robin => Batman's sidekick

In the face of this, our arbitrary choice was to consider the Proper Noun as the "head" of the relation, that is, the "defined expression" in the terms of [7].

3 Techniques

In our appositive extraction approaches, we use two basic machine learning techniques: Hidden Markov Models (HMM) and Transformation Based Learning (TBL).

3.1 Hidden Markov Models

Hidden Markov Models [8] technique is a powerful probabilistic tool to model sequential data. HMM has been the focus of many applied work, such as part-of-speech (POS) tagging, text segmentation and voice recognition.

HMM provides the modeler with two basic concepts: observations and *hidden* states. In text mining applications, a sequence of words is the observed data, whereas the hidden states typically model structural or semantic information associated to the given text. The HMM parameters are set to maximize the likelihood between what we are observing and what we are trying to discover. Hidden Markov Models are frequently used when we want to model the probability of a sequence of events. They are represented by a nondeterministic finite automaton with three kinds of probabilities: the initial, the transition and the emission probabilities. The main difference between HMM and ordinary Markov chains is that the state sequence of the automaton or its deterministic function is not observed, just some probabilistic function of it.

3.2 Transformation Based Learning

Transformation Based error-driven Learning is a successful symbolic machine learning method, introduced by Eric Brill [9]. It has since been used for several linguistic tasks, such as part-of-speech (POS) tagging, parsing, prepositional phrase attachment and phrase chunking, achieving state-of-the-art performance in many of them.

The central idea of the TBL algorithm is to generate an ordered list of rules that correct tagging mistakes in the corpus, which have been produced by an initial guess. The rules are generated according to a set of rule templates, which are meant to capture the relevant feature combinations.

The learning algorithm is a mistake-driven greedy procedure that iteratively acquires a set of transformation rules. The TBL algorithm can be depicted as follows:

1. Starts applying a initial guess classification for a un-annotated version of the training corpus;
2. Compares the resulting classification with the correct one and, whenever a classification error is found, all the rules that can correct it are generated by instantiating the templates with the current token's context. Normally, a new rule will correct tagging errors, but will also generate some other errors by changing correctly tagged tokens;
3. Computes the rules' scores (errors repaired - errors created). If there isn't a rule with a score above an arbitrary threshold, the learning process is stopped;
4. Selects the best scoring rule, stores it in the set of learned rules and apply it to the corpus;
5. Returns to the step 2.

4 Modelling

4.1 Corpus

The corpus used in this study contains 4,638 sentences and was derived as follows:

- the SNR-CLIC corpus described in [10], which is annotated with pos and noun phrase tags;
- newswire texts, annotated with POS tags using the Lacio-Web Tagset, and with NP tags using the SNr identifier [11].

The appositive relation is encoded using the tag set {E+, E−, E+−, A+ , A−, O}, henceforth called the EA tag set. Here, the E tag is associated to words in an entity description, the A tag is associated to words in an appositive related to an entity, and the O tag is associated to words not included in a appositive relation. The minus sign (−) indicates that the other member of the relation is on the left, and the plus sign (+) indicates that the other member is on the right. The E+− tag is used when an entity has two appositions related to it, one to the left and another to the right. An example of the encoding is shown below.

- The/O explanation/O is/O from/O Luis/E+ Eduardo/E+ ,/O Association's/A- president/A- ./O

The appositive annotation was semi-automatic, combining the application of the Baseline System described in 4.2 and manual review. The resulting corpus contains 583 examples of the relation.

4.2 Baseline System

The Baseline System (BLS) is a set of TBL rules manually created. The aim of this BLS is to extract appositive relations which have simple patterns. The considered patterns are listed below:

Cases having one appositive related to an ENTity.

- NP , ENT ,
- ENT , NP ,
- NP (x), ENT ,
- ENT (x) , NP ,

Cases having two appositives related to the same entity.

- NP , ENT , NP ,
- ENT, NP and NP ,

Where:

- ENT = Entity description, composed by a noun phrase with the form: [Article][Adjective][Noun] Proper Name, where the article, the adjective and the noun are optional;
- NP = a noun phrase;
- (x) = sequence of words in parenthesis.
- and = the coordinated term "and" ("e" in Portuguese).

The final comma could also be changed by ".", "(" or ":".

4.3 HMM Model for Appositive Extraction

One quick way to model appositive extraction is to use the EA−tags (E+, E−, E+−, A+, A−, O) as the hidden states and the pos-tags as the emissions. Each sentence to be classified is mapped to its pos-tag sequence . What we want to find is the corresponding most likely sequence of EA-tags. The HMM probabilities are estimated as normalized frequencies obtained by feature counting in the training data.

This simple model is very inefficient, since it has just a small number of states and it does not take advantage of the inherent structure of the noun phrases neighborhood.

To overcome this limitation we introduced new enhanced states. They were generated online and based on the manual labeling. The new tags are:

- OAT, a tag immediately after a given T tag;
- OBT, a tag immediately before a given T tag;
- OCT, a tag immediately after a OAT tag;
- ODT, a tag immediately before a OBT tag;
- OET, a tag immediately before and after a given T tag;
- OHT, a tag immediately after a given T tag and before another T tag;
- 1[+|-]T, the first T[+|-] tag of a sequence;
- and 0[+|-]T the last T[+|-] tag of a sequence.

where T represents one of the possible EA−tags. For instance, the E+ tag generates the tags: {OAE, OBE, OCE, ODE, OEE, OHE, 1+E and 0+E}.

To avoid mapping a tag to two or more different states, we added an extra relabeling procedure, which uses a previous established order of preference for the states. An example of this is shown in Table 1.

Table 1. Relabel procedure for HMM state generation

word	pos-tag	np-tag	ea-tag	hmm state
...				
o	ART	I	O	O
presidente	N	I	O	O
se	PROPESS	O	O	O
reuniu	V	O	O	ODA
com	PREP	O	O	OBA
o	ART	I	A+	1+A
procurador-geral	N	I	A+	0+A
,	,	O	O	OHA
Aristides=Junqueira	NPROP	I	E-	1-E
,	,	O	O	OAE
para	PREP	O	O	OCE
discutir	V	O	O	O
...				

With this effective relabeling procedure we enhance our results, as a consequence of the *O* EA-tag refining. Now we can improve our model by taking advantage of the available lexical information. In HMM appositive extraction, a preposition can lead to many errors if treated by its pos-tag. Whenever prepositions appear in a sentence, we replaced them by their corresponding lexical item. We also use a small list of words like 'que' as observations.

The HMM model often makes some simple tagging mistakes. These mistakes can be easily corrected by a set of fixed rules that are applied after the classification of the sequence. They were obtained by a set of TBL rules and evaluated with the corpus. We kept only the rules that showed a positive score.

To help the classification process, a baseline system (BLS) can be evaluated before the HMM classification and its non null evaluated EA-tags used as observations for the HMM model.

4.4 TBL Model for Appositive Extraction

The following TBL configuration was used:

Initial Classification: two different initial classifications were tested: (1) The Baseline System, and (2) the HMM Model, both described previously.

Templates: we tested various templates that combined the features *word*, *pos*, *np* and *EA-tags*, using a context window with 12 tokens of size: 6 tokens to the left, the token currently being corrected and 5 tokens to the right. The template set that provides our best results contains only 8 templates.

5 Experimental Results

Each experiment was realized using cross-validation with 10 sampling. For each sampling, the corpus was randomly divided into 70% for training and 30% for test. We made five experiments:

1 BLS: the application of the Baseline System
2 Plain HMM: HMM with the enhanced states
3 BLS + HMM + rules: HMM with Baseline System as Initial Classification, followed by a set of fixed rules
4 HMM + TBL: TBL with Plain HMM as Initial Classification
5 BLS + TBL: TBL with Baseline System as Initial Classification

Table 2 shows the results for each experiment.

Table 2. Comparison between the 5 experiments and the parser PALAVRAS

	Precision (%)			Recall (%)			F-score (%)		
Experiment	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
BLS	70.34	76.59	67.29	**81.61**	86.67	73.49	75.52	79.19	70.32
Plain HMM	50.23	55.20	46.18	67.13	71.25	62.71	57.41	60.45	54.50
BLS + HMM + rules	**77.57**	79.78	74.19	69.32	74.37	64.89	73.19	76.68	70.11
HMM + TBL	55.26	73.45	43.94	34.70	45.86	29.53	42.49	56.46	37.38
BLS + TBL	74.69	78.26	71.02	78.67	81.82	74.73	**76.61**	80.00	73.00
PALAVRAS	80.01			57.73			67.10		

An error analysis of the experiment BLS+TBL, which achived the best F-score, showed that the difference between adverbial and appositive structures continues a problem when adverbial adjuncts are dislocated. Another difficulty concerns the correct identification of the reference of the apposition when the NP has more than one noun. For instance, in a sentence like *The proposal of the president, Lula, was...*, the correct relation is *Lula => president* but the system extracted *Lula => The proposal of the president*. In some cases, however, the sentence is highly ambiguous itself, and identifying the correct referent is difficult even for humans, as in *The son of the president, Lula, was....* Finally, some highly complex structures, such as *The death of Vincent Foster, advisor and friend of the president of the US, Bill Clinton, caused...* which presents an *embedded apposition*, were difficult do identify as well. In a pos-processing phase, we applied a very simple heuristic to the extracted relations in order to obtain, when it was possible, appositives whose head were a single NP. Fig. 1 shows some of the extracted relations.

It is important to note that, for the TBL extractor, it would be hard to achieve a 100% precision score. This limitation results from the NP identification task, which is prior to appositive identification. The experiments here used the SNRidentifier [11], a NP identifier that recognizes NPs characterized by having exactly one nominal head, precluding pronominal NPs and coordinated NPs

Head	Definition
Mike Tyson	ex-campeão mundial dos pesos-pesados
Damasco	capital Síria
Fernando Henrique Cardoso	candidato de o PSDB ministro de a Fazenda presidente eleito
Zuenir Ventura	jornalista autor do livro "1968"

Fig. 1. Some relations identified by the TBL-based extractor

[10]. As a result, appositives with embedded clauses (*The hotel that will host the conference, Great Hotel, accepts...*), and appositives whose head is elliptical (*The president of the Senate, Humberto Lucena, and of the Parliament, Inocêncio de Oliveira, will meet...*) were not considered.

The final step of the experiments was a comparison between our results and those obtained with the parser PALAVRAS. As the output of PALAVRAS is completely different from ours, we randomly selected 317 sentences to be parsed, and the results were manually reviewed. As can be seen from Table 2, TBL and HMM extractors performed better than PALAVRAS. However, the comparison between the machine learning outputs and PALAVRAS, although illustrative, is not really informative, for PALAVRAS considers appositives only a chunk that doesn't belong to a relationship. So it computes the presence of an apposition, but it doesn't inform to which NP the appositive is related. We then adjust the TBL and HMM outputs to facilitate the comparison. The new output only indicates the presence of an appositive, instead of identifying a relation between two NPs. The new results are presented in Table 3. We can note that the performance of both TBL and HMM, higher than those of Table 2, indicates that identifying the relation is a more complex task.

Table 3. Results of comparable outputs between the ML approaches and the PALAVRAS

Experiment	Precision (%)	Recall (%)	F-score (%)
PALAVRAS	80.01	57.73	67.10
BLS + HMM + rules	**84.63**	80.12	82.27
BLS + TBL	78.79	**87.84**	**83.03**

6 Concluding Remarks

In this paper, we presented five experiments addressing the identification of appositives in Portuguese unrestricted texts. As far as we know, this is the first machine-learning approach to appositive identification. In previous works [1,2,3,4,5], this task has been addressed using full parsing or specific heuristics.

The results obtained here are encouraging. Our TBL-based system produced the best one, a 76.61% F-score. This is a significant increase when compared to PALAVRAS, the only tool that identifies appositives for the Portuguese language, whose corresponding F-score is 67.10%.

The results obtained with TBL can be enhanced, since some mistakes produced by the TBL-based extractor were caused by the underlying NP model. An alternative would be to adjust the NP model for this task. On the other hand, Active Learning can be used to incrementally increase the training corpus with more relevant examples.

References

1. Phillips, W., Riloff, E.: Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In: Proceedings of the EMNLP. (2002)
2. Caraballo, S.: Automatic construction of a hypernym-labeled noun hierarchy from text (1999)
3. Cardie, C., Wagstaff, K.: Noun phrase coreference as clustering. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, University of Maryland, MD, Association for Computational Linguistics (1999) 82–89
4. Soon, W., Ng, H., Lim, D.: A machine learning approach to coreference resolution of noun phrases (2001)
5. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. (2002) 104–111
6. Bick, E.: The Parsing System Palavras: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. PhD thesis, Aarhus University (2000)
7. Quirk, R., Greenbaum, S., Leech, G., Svartvick, J.: A Comprehensive Grammar of the English Language. Longman (1985)
8. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. In: Proc. of IEEE. (1989) 257–286
9. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics **21** (1995) 543–565
10. Freitas, M.C., Garro, M., Oliveira, C., Santos, C.N., Silveira, M.: A anotao de um corpus para o aprendizado supervisionado de um modelo de sn. In: Proceedings of the III TIL / XXV Congresso da SBC, So Leopoldo - RS (2005)
11. Santos, C.N.: Aprendizado de mquina na identificao de sintagmas nominais: o caso do portugus brasileiro. Master's thesis, IME, Rio de Janeiro - RJ (2005)

Parameterized Imprecise Classification: Elicitation and Assessment

Isabela Drummond[1] and Sandra Sandri[2]

[1] Instituto Nacional de Pesquisas Espaciais, S.J.Campos - SP 12227-010, Brasil
isabela@lac.inpe.br
[2] Campus Univ. Automata de Barcelona, Bellaterra 08193, Spain
sandri@iiia.csic.es

Abstract. This work is based on classifiers that can yield possibilistic valuations as output. The valuations may have been obtained from a labeled data set either directly as such, by possibilistic classifiers, by transforming the output of probabilistic classifiers or else by adapting prototype-based classifiers in general. Imprecise classifications are elicited from the possibilistic valuations by varying a parameter that makes the overall classification become more or less precise. We introduce some indices to assess the accuracy of the parameterized imprecise classifications and their reliability, thus allowing the user to choose the most suitable level of imprecision and/or uncertainty for a given application.

1 Introduction

Given a set of classes C and a set of objects Z that one is interested to classify, a classification system usually assigns a single class $c \in C$ to each element of $z \in Z$, eventually relying in randomness to break ties. Some of these systems are also capable of yielding a valuation in a given uncertainty model quantifying the evidence in favor of each class, e.g., "there is a probability of .3 that z_0 belongs to class c_1 and .7 that it belongs to c_2". One can also think of classifiers that yield an order of preference on the set of classes, e.g. "c_1 is better than c_2 as the class of z" or simply a subset of C, e.g. "the class of z_0 is either c_1 or c_2".

When the classifier yields precise results, the usual means to assess the quality of these results is also very precise: the rate between the number of objects correctly classified and the total number of objects, called the classifier accuracy, or its corresponding error rate.

In any case, the accuracy rate obtained for the set used to train a classifier can be used as an estimation of the real accuracy that will be obtained once the objects of interest are presented to the trained classifier.

However, knowing that a certain amount of objects are expected to be ill-classified give us no clues about which is their identity. In applications in which a wrong classification may cause a great loss, it is often preferable to obtain a to obtain a complete imprecise response (equivalent to *"I don't know"*) than to have a high risk of getting a precise but incorrect answer. In such a case, it is better to have a way for the system to express that an object is possibly an outlier, in other words that there is not

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 319–328, 2006.

enough evidence to classify it one way or another. In other words, it may be more advantageous to trade uncertainty for imprecision. In this work, we propose the use of a parameterized family of imprecise classifiers and a set of indices to help the decision maker to choose one of them: the parameter creates a series of imprecise classifiers, varying from accurate and reliable but imprecise to more precise but less accurate and less reliable.

Our work is focused on discriminant functions that are normalized possibility distributions [5]. From these distributions one derives a valuation containing a (non-necessarily nomalized) possibility distribution on the potential classification of a given element of interest. Many classifiers have discriminant functions from which a possibility distribution can be obtained (see [7] for a review on fuzzy classifiers). The approach can be used for classifiers that yield probability distributions on the set of classes, applying a transformation that still produce a possibility distribution. Last but not least, the approach can be used on any classifier based on a set of prototypes (e.g. obtained by clusterizing the training set) by creating the possibilistic distributions as a function of the Euclidean distance to the prototypes (e.g. using a k-nn algorithm).

This work is organized as follows. In Section 2, we first discuss imprecise classifiers and then present some indices to assess the accuracy of their outputs. In Section 3 we show how to elicit a parameterized family of imprecise classifiers from possibilistic ones. In Section 4 we propose indices to assess the reliability of a valuation-based imprecise classifiers, that take into account the confidence of the imprecise classifier on its outputs. In Section 5 we compare two basic classifiers and the imprecise classifiers derived from them and Section 6 brings the conclusion.

2 Assessing Imprecise Classification Accuracy

Let $C = \{c_1, ..., c_m\}$ be a set of m distinct classes. Let Z be a labeled data set: each z is a vector of size n, whose value z_i in the i-th position is related to an attribute A_i defined over a domain Ω_i. For simplicity, we assume $\forall i, \Omega_i = R$. The class label of z is denoted by $l(z) \in C$ or, whenever possible, simply by $c^* \in C$.

A classifier is defined as a mapping $D : R^n \rightarrow C$ [8]. In the canonical model, there exist m discriminant functions $g_i : R^n \rightarrow R$, each of which computing a score relative to class c_i. Usually, $\forall z \in R^n, D(z) = \arg\max g_i(z)$, i.e., the classifier assigns z to the class with the highest score and ties are broken randomly. Here we shall call $valuation$ to the distribution $v(z) : C \rightarrow R$, that contains the class scores calculated for a given point z, defined as $v(z)(c_i) = g_i(z)$.

Here we address classifiers that assign a $subset$ of C to any given $z \in R^n$. An *imprecise classifier* is a mapping $Cl : R^n \rightarrow 2^C$: when $|Cl(z)| = 1$, we say the class is $pure$, and when $|Cl(z)| > 1$ we say the class is $compound$ [1]. In other words, an imprecise classifier is a classifier that yields the class of z as a crisp subset of the total set of classes C.

To illustrate this work, we have taken a data set from [7], containing 800 points divided in 3 classes. We used a set of 400 points as the training set and the remaining as test set. Figure 1 brings a comparison between two classifiers Cl_1 and $Cl_{.4}$. The

[1] In tree classifiers, compound classes are usually known as $impure$ [8].

the points correspond to elements in pure classes, the light (respec. dark) ones indicate correct (respec. incorrect) classification; the (light) circles correspond to elements in compound (but nor completely imprecise) classes and the (dark) triangles indicate elements with completely imprecise classification.

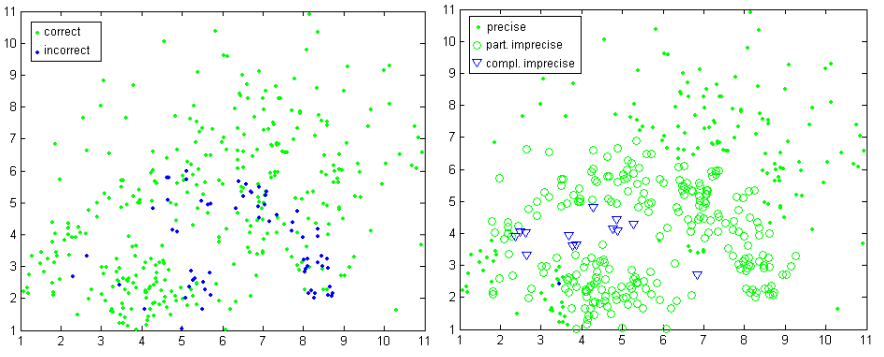


Fig. 1. A precise (Cl_1) and an imprecise ($Cl_{.4}$) classification for the cone-torus training data [7] (prototype configuration $< 5, 5, 10 >$)

A confusion matrix brings a summarization of the results obtained by a classification: each entry $a_{i,j}$ denotes the number of elements in a data set that belong to class c_i that have been classified as belonging to class c_j. For an imprecise classification we need an imprecise confusion matrix; we still have m rows, one for each class in c, but 2^m columns, one for each set in the powerset of C, including the empty set. Table 1 brings two examples of an imprecise confusion matrix; the one from classifier Cl_1 can be put in the form of a "precise" confusion matrix, since it corresponds to a precise classification, whereas the one from classifier $Cl_{.4}$ cannot.

Let $c_{j...k}$ denote the set $\{c_j, ..., c_k\} \subseteq C$ and let us denote an entry on the row corresponding to class c_i and classification $Cl(z) = \{c_j, ..., c_k\}$ as $a_{i,j...k}$. Entries $a_{i,i}, i = 1, m$ thus contain the number of elements that are perfectly classified: the classification is both correct and precise. For class c_3 in the table for $Cl_{.4}$ we see, for instance, that 115 elements in the data set are correctly, and precisely, classified as belonging to c_3 and that 1 element was incorrectly classified as belonging to c_1. The entries $a_{i,j}, i \neq j$ bring the errors that correspond to what is called $overconfidence$ in the expert judgement literature [2,9]: the assessment is as precise as possible but completely wrong. We see also that 12 elements of c_3 are correctly, albeit imprecisely, classified as belonging to either class c_1 or c_3.

The last column corresponds to $Cl(z) = C$ and brings the number of elements whose classification is correct but completely imprecise. The first column, $\emptyset$, brings the number of elements for which the classifier could not assign any classification. Note that the first and last columns are not exactly the same: the first one models total conflict, whereas the last one models equal evidence for all possibilities.

The data set Z can be partitioned according to correctness and cardinality. Given a classifier Cl, we denote by Z^{*k} the set of elements in Z that are correctly classified as

Table 1. Example of imprecise confusion matrices for classifiers: a) precise and b) imprecise

a)

Cl_1	$\emptyset$	c_1	c_2	c_3	c_{12}	c_{13}	c_{23}	c_{123}
c_1	0	83	2	7	0	0	0	0
c_2	0	0	60	39	0	0	0	0
c_3	0	8	8	193	0	0	0	0

b)

$Cl_{.4}$	$\emptyset$	c_1	c_2	c_3	c_{12}	c_{13}	c_{23}	c_{123}
c_1	0	19	0	0	6	63	0	4
c_2	0	0	14	0	4	0	78	3
c_3	0	1	0	115	0	12	75	6

belonging to a class of cardinality k; by Z^* the set of elements Z whose classification is neither incorrect nor completely imprecise,and by Z_{err} (respec. by Z_{imp}) the set of elements incorrectly classified (respec. with completely imprecise classification):

- $Z^{*k} = \{z \in Z^* \mid |Cl(z)| = k\}$
- $Z^* = \{z \in Z \mid c^* \in Cl(z) \wedge Cl(z) \neq C\} = \bigcup_{i=1,m-1} Z^{*k}$
- $Z_{err} = \{z \in Z \mid c^* \notin Cl(z)\}$
- $Z_{imp} = \{z \in Z \mid Cl(z) = C\}$

It is easy to see that Z^*, Z_{imp} and Z_{err} are mutually exclusive and that $Z = Z^* \cup Z_{imp} \cup Z_{err}$. In Figure 1, the green (respec. blue) points represent elements of Z^{*1} (respec. Z_{err}), the circles represent elements of Z^{*2} and the triangles represent the elements of Z_{imp}.

Usually, the quality of a precise classifier D in relation to a data set Z is assessed by its $accuracy$, taken as the proportion of elements correctly classified in Z:

$$acc(D, Z) = |\{z \in Z \mid l(z) = D(z)\}|/|Z|. \tag{1}$$

In tree-classifiers, the indices used to qualify the $impurity$ of a result is usually assessed by Shannon's entropy or Gini diversity index [11]. However, in the case of imprecise classifiers we need indices that take into account the accuracy of the imprecise classification, which is not possible using those indices.

To verify the quality of an imprecise classifier, we define accuracy indices $qual^{*k}$ and $qual^*$, based on Z^{*k} and Z^*. We also define indices err and imp, that measure error and complete imprecision.

- $qual^{*k}(Cl, Z) = |Z^{*k}|/|Z|$
- $qual^*(Cl, Z) = |Z^*|/|Z|$
- $err = |Z_{err}|/|Z|$
- $imp = |Z_{imp}|/|Z|$

In the example depicted in Figure 1, we have $qual^{*1}(Cl_1, Z) = qual^*(Cl_1, Z) = 336/400 = .84$, $qual^{*1}(Cl_{.4}, Z) = 148/400 = .37$ and $qual^*(Cl_{.4}, Z) = 386/400 = .965$. We see that with imprecise classification we trade precision for accuracy, in the sense that even though the classifier's output is less precise, it commits less errors.

3 Parameterized Imprecise Classifiers

In the previous section, we focused on assessing the quality of classifiers that simply yield a subset (a singleton or not) as output. In this section we present a framework for

the elicitation and (quality) assessment of imprecise classifiers, that are derived from classifiers that yield a possibilistic valuation as result.

Since from a probabilistic valuation one can always derive a possibilistic one (see [6]), the framework can be used straightforwardly in probabilistic classifiers. Yet more generally, the framework can be used on any set of data that has a set of prototypes associated to each class. Using the prototypes one can generate either probabilistic (e.g. using k-nn classification and a metric on the attribute space such as the Euclidean-distance) or a possibilistic classifier such as presented in [10] or [3], based on similarity-relations.

Our work is focused on discriminant functions that are (normalized) possibility distributions: each g_j is a mapping $g_i : R^n \rightarrow [0,1]$, such that $\exists z \in R^n, g_i = 1$, i.e., each class has at least one element on the attribute space that is completely compatible with it. The possibilistic valuation $v(z) : C \rightarrow [0,1]$ associated to z is not necessarily a normalized possibility distribution, i.e., there may not exist a class with which z is completely compatible. In this work, we denote by π the possibility distribution obtained by normalizing valuation v, calculated as $\pi(z)(c_i) = v(z)(c_i)/K$, where $K = max_i v(z)(c_i)$.

When a classifier yields a valuation, it usually produces as output the class that has highest support in the valuation. Let us call the procedure that chooses the (usually precise) output out of a valuation as the classifier's decision-maker. What we propose here is to take one such classifier and create a parameterized family of decision-makers. Given a valuation $\pi(z)$ yielded by a classifier Cl to an element $z \in Z$ we create a decision-maker $Cl_\alpha, \alpha \in]0,1]$, that yields as output an imprecise class for z as

$$Cl_\alpha(z) = \{c \in C \mid \pi(z)(c) \geq \alpha\} \tag{2}$$

i.e. the class of z is given by the level-cut with degree α [5].

We applied fuzzy c-means clustering [1] on the cone-torus data set [7] and generated possibilistic valuations following the similarity-relations approach proposed in [3,4]. In this approach, for each class c_i we use t_i prototypes; each prototype configuration is denoted by $< t_1, ..., t_n >$. The classifications depicted in Figure 1 are from Cl_1 (precise) and $Cl_{.4}$ (imprecise) based on a possibilistic classifier Cl with a $< 5, 5, 10 >$ prototype configuration. Figure 2.a) brings the imprecision-based indices $qual^*$, $qual^{*1}$ and $qual^{*2}$ obtained from classifier Cl, varying $\alpha \in]0,1]$, thus between the most imprecise classification and the most precise one that we can obtain with that configuration.

Note that, for any Z, $qual^*(Cl_1, Z) = qual^{*1}(Cl_1, Z)$ and $qual^{*k}(Cl_1, Z) = 0, \forall k \neq 1$ (see point a in Figure 2). Value b is related with the number of points that are perfectly classified no matter the value of α, i.e., those points for which the possibilistic valuation is a precise fuzzy set. Value c is the lowest value of α in which all correctly classified points in the data set become useful, i.e. their classification is correct but not completely imprecise; d is the value of α that yields the highest value for $qual^*$ and e indicates the lowest value α for which Cl_α produces incorrect classifications. These seem to be good parameters to base an imprecise classification. In Figure 2, $a = .84$, $b = .1775$, $c = .76$, $d = .52$ and $e = .38$.

4 Assessing Imprecise Classification Reliability

Let us suppose that $\pi(c), c \in C$, models the possibility that the correct class of a point z is c, i.e. that $c^* = c$. Several uncertainty measures can be calculated from a normalized possibility distribution $\pi(.)$ to measure the confidence that c^* belongs to a set $A \subseteq C$, such as

- $\Pi(A) = max_{u \in A} \pi(u)$
- $N(A) = 1 - \Pi(\overline{A}) = min_{u \notin A} 1 - \pi(u)$
- $N\Pi(A) = (\Pi(A) + N(A))/2$
- $P(A) = \sum_{u \in A} p(u)$.

where Π and N are a possibility and a necessity measures respectively [5], and $N\Pi$ is their average; P is a probability measure derived from a probability distribution p that can be obtained by the application of a transformation on π [6][2]. When we make $A = Cl(z)$ these measures can be seen as confidence measures of the classifier about its own output.

For a given imprecise classifier Cl_α, we define the individual accuracy of its classification of point z as

$$acc(Cl_\alpha, z) = \begin{cases} 1, \text{ if } l(z) \in Cl_\alpha(z) \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

Taking $\pi(z)(.)$, the individual uncertainty-based accuracy indices Π_acc, N_acc, P_acc and $N\Pi$_acc are defined through the use of

$$f\_acc(Cl_\alpha, z) = f(Cl(z)) \times acc(Cl_\alpha, z) \tag{4}$$

by substituting f with Π, N, P and $N\Pi$, respectively.

Note that whenever $\alpha \geq \beta$ we have $Cl_\alpha(z) \subseteq Cl_\beta(z)$ and thus $acc(Cl_\alpha, z) \leq acc(Cl_\alpha, z)$, $f(Cl_\alpha(z)) \leq f(Cl_\beta(z))$ and $f\_acc(Cl_\alpha, z) \leq f\_acc(Cl_\alpha, z)$.

Similarly, given a data set Z, we obtain its overall Π-, N-, P- and $N\Pi$- uncertainty-based accuracy indices, respectively through the use of[3]

$$f\_acc(Cl_\alpha, Z) = \sum_{z \in Z} f\_acc(Cl_\alpha, z)/|Z| \tag{5}$$

by substituting f with Π, N, P and $N\Pi$, respectively. Note that $N\_acc(Cl_\alpha, Z) \leq P\_acc(Cl_\alpha, Z), N\Pi\_acc(Cl_\alpha, Z) \leq \Pi\_acc(Cl_\alpha, Z), \forall \alpha$. These indices can be taken as the $reliability$ of the imprecise classifier.

In Figure 2.b) we show the reliability indices relative to classifiers $Cl_\alpha, \alpha \in]0, 1]$, obtained from a possibilistic classifier Cl with a $< 5, 5, 10 >$ prototype configuration; the transformation from possibilistic valuation to probabilistic one in order to obtain P_acc was done as proposed in [6].

[2] Note that if we were using valuations provided by a probabilistic classifier, we would use p directly to obtain P but would need to transform p into π to obtain Π and N.

[3] Here we are using the same symbol for individual and overall uncertainty-based indices so not to complicate notation any further.

Value u in Figure 2 is related to the probability measures obtained for the precise classification, i.e. $P\_acc(Cl_1, Z)$. In our example, the confidence, measured as a probability, that $l(z)$ belongs to $Cl(z)$ is of $P\_acc(Cl_1, Z) = .6483$ in average. Using $Cl_{.76}$ the average raises to .8523. As expected, as imprecision raises, so raises the confidence in an output. Among the uncertainty-based accuracy indices, the probabilistic one seems to be a good choice, since probability is a more straightforward concept and it is neither too optimistic as Π_acc nor as pessimistic as N_acc. On the other hand, $N\Pi\_acc$ can be a good substitute for P_acc if one wants to avoid transforming the valuations from one uncertainty framework to another.

Note that even though P_acc and $N\Pi\_acc$ overlap most of the time in our example, theoretically they are not necessarily the same. However, $N(A) \leq P(A) \leq \Pi(A)$ for acceptable possibility-probability transformations and $N(A) \leq N\Pi(A) \leq \Pi(A)$ by definition. Consequently, $P\_acc(A)$ and $N\Pi\_acc(A)$ are both bounded by $N\_acc(A)$ and $\Pi\_acc(A)$, and are thus somewhat similar measures. Moreover, whenever $P\_acc(., z) \in \{0, 1\}$ we have $N\Pi\_acc(., z) = P_{acc}(., z)$. Thus for the extremely imprecise classifiers we should have the $N\Pi\_acc(., Z)$ very close to $P\_acc(., Z)$; the same behavior should occur with extremely good classifiers, i.e. those which are both precise and accurate.

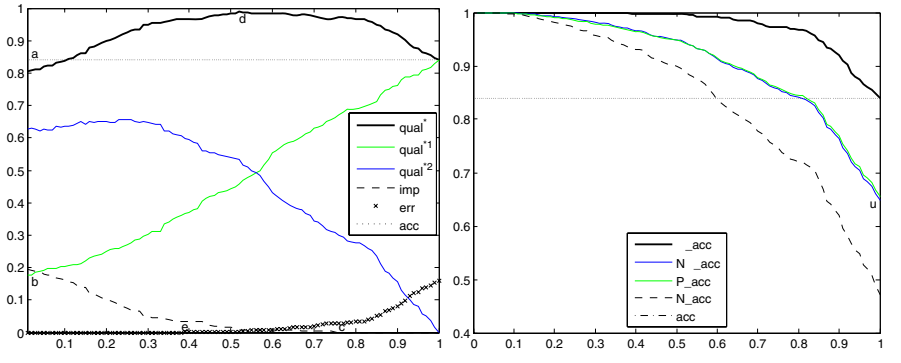


Fig. 2. Performance indices of Cl_α for training data (config. $< 5, 5, 10 >$): a) imprecision-based, b) uncertainty-based

In Figures 3 and 4 we show the results of applying the same classifiers presented previously but on a test data also consisting of 400 points. The data is synthetic and the training and test data sets follows the same distributions. We see that in this case, as expected, the indices for the test data set behave closely to those of the training data set. With real data, one cannot expect the results to be so related.

Aamof, with real data, we need a leap of faith to use the indices obtained from the training data to guide us to classify the data of interest. However, that happens for any algorithm that uses parameters derived by learning from examples. If the training data reflects the overall behavior of all the data, then the results become more trustful.

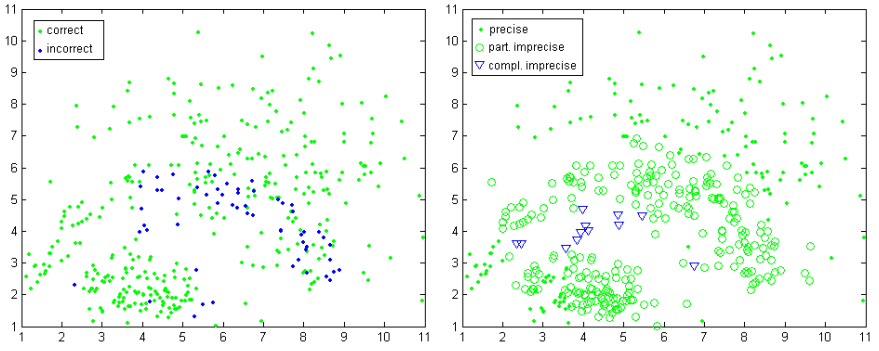


Fig. 3. Precise (Cl_1) and imprecise ($Cl_{.4}$) classification for test data (config. $< 5, 5, 10 >$)

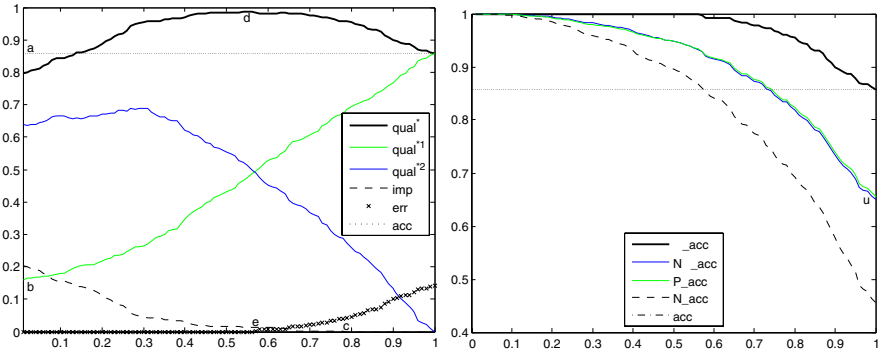


Fig. 4. Performance indices of Cl_α for test data (config. $< 5, 5, 10 >$): a) imprecision-based, b) uncertainty-based

5 Comparing Distinct Parameterized Imprecise Classifiers

Our main goal here is to allow a decision-maker to get the best trade-off between precision and accuracy from a possibilitic classifier. However, the formalism may be used to compare two distinct classifiers. In the previous sections we illustrated our work with a classifier with prototype configuration $< 5, 5, 10 >$. In Figures 5 and 6 we show the results of applying the formalism with prototype configuration $< 2, 2, 4 >$. The results for the test data sets are similar to the ones shown for the training set. The higher quality of classifier $< 5, 5, 10 >$ over classifier $< 2, 2, 4 >$ can be seen by a visual comparison of Figures 1 and 5. But, what is more interesting, that superiority can be seen by comparing Figures 2 and 6.

An ideal classifier is such that, for any α, the value of $qual^*$ is close to 1, the values of err and imp are close to 0 and the reliability indices are close to 1. We see that, for no matter which α, the classification using the $< 5, 5, 10 >$ configuration yields a higher proportion of correct classification, either precise or partially imprecise, than

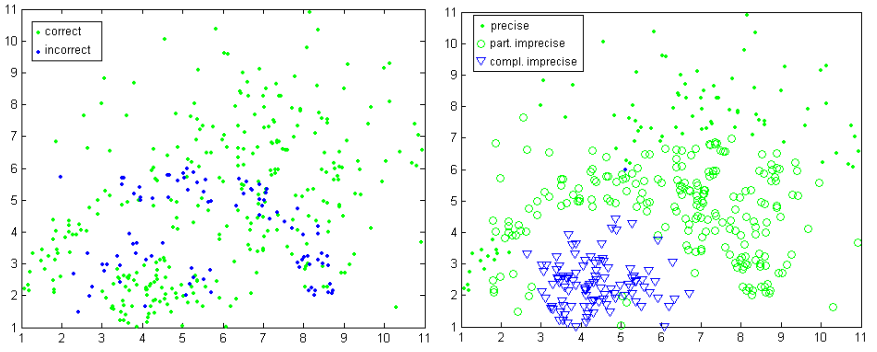


Fig. 5. Precise (Cl_1) and imprecise ($Cl_{.4}$) classification for training data (config. $< 2, 2, 4 >$)

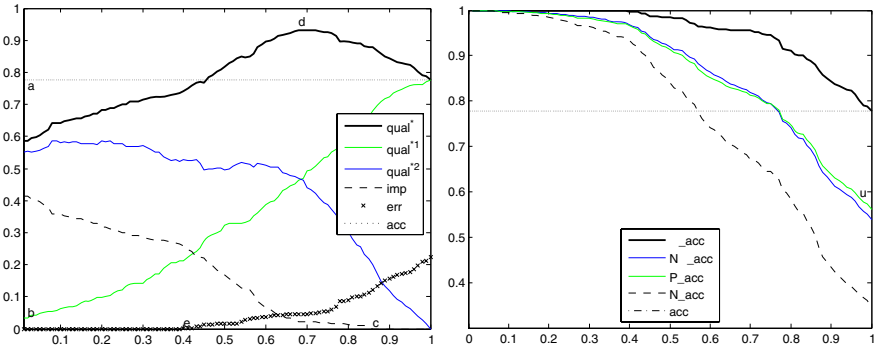


Fig. 6. Performance indices of Cl_α for training data (config. $< 2, 2, 4 >$): a) imprecision-based, b) uncertainty-based

those yielded by $< 2, 2, 4 >$, the proportions of either completely imprecise or incorrect classification are also lower for $< 5, 5, 10 >$, and $< 5, 5, 10 >$ is more reliable than $< 2, 2, 4 >$. That illustrates how the proposed indices reflect the quality of the classifiers. This is specially interesting in applications where n (the number of attributes describing the data) is higher than 3, for which straightforward visualization is not possible.

6 Conclusion

We have proposed a parameterized imprecise classification framework and presented some indices to assess the resulting imprecise classifiers, such that the most adequate trade-off between accuracy and precision can be chosen according to the decision-maker's needs.

Here we have illustrated our approach using very small variations in the level of imprecision parameter. However, since the $qual^*$ curve is either unimodal or has an

interval as mode, one can find the optimal value for the parameter using a small number of choices for the parameter, thus increasing efficiency.

The confusion matrices employed here may have $2^{|C|}$ columns and the approach could thus be potentially unfeasible for large values of C. However, using prototype-based classifiers, the complexity of the approach is in fact bounded by $|C| \times |Z| \times s$, where s is the largest number of prototypes used for a class, instead of $|C| \times |Z|$ as in the case of precise classification.

The quality indices proposed do not put together imprecision and accuracy, as done by other approaches such as in [9]. Here, an accuracy and a reliability index is given for any given level of imprecision. In the future we intend to study the combination of these indices, also according to the level of imprecision. We also intend to extend the approach to consider the reliability concerning the individual classes or the union of some of them.

We have not addressed the question of combining imprecise classifiers. The combination of classifiers is usually called ensemble, and can be done using the classifer's "final" results, i.e. the precise classification, or, in the case of valuation-based classifiers, inside the valuation uncertainty model (see [7],[8] for a through study in both themes). We envisage to address this theme in future work. Also, we intend in the future to include the means to assess imprecise classifications taking cost (or gain) into account.

Acknowledgements

Isabela Drummond acknowledges support from the Brazilian funding agency CAPES.

References

1. J. Bezdek, R. Ehrlich and W. Full, FCM: The fuzzy c-means algorithm, *Computers & Geosciences* **10, 2-3** (1984), 491–263.
2. R. M. Cooke, *Experts in uncertainty*, Oxford University Press, 1991.
3. I. Drummond and S. Sandri, A clustering-based possibilistic method for image classification, *Advances in Artificial Intelligence*, LNAI **3171**, Springer Verlag (2004), 454–463.
4. I. Drummond and S. Sandri, A clustering-based fuzzy classifier, in *Vuité Congrés Catalá d'Intel.ligência Artificial (CCIA2005)*, October 2005, Sardegna, Italy. Artificial Intelligence Research and Development and Frontiers in Artificial Intelligence and Application. Berlin: IOS Press, 2005, v.131, p. 247-254.
5. D. Dubois and H. Prade, *Possibility Theory*, Plenum Press, 1988.
6. D. Dubois, H. Prade and S. Sandri, On possibility/probability transformations, in *Fuzzy logic:state of the art* R. Lowen and M. Roubens, Eds, Kluwer (1993), 103–112.
7. L. I. Kuncheva, *Fuzzy classifier design*, Springer Verlag, 2000.
8. L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*, Wiley, 2004.
9. S. Sandri, D. Dubois and H. Kalfsbeek, Elicitation, assessment and pooling of expert judgments using possibility theory, *IEEE Transactions on Fuzzy Systems* (1995), 3(3), 313–335.
10. J. M. Keller, M. R. Gray and J.A. Givens A fuzzy k-nearest neighbor algorithm, in *IEEE Transactions on Systems, Man and Cybernetics*, (1985), 15(4), 580–585.
11. J.C. Bezdek , J. Keller, R. Krisnapuram and N. R. Pal *Fuzzy models and algorithms for pattern recognition and image processing*, Springer, 1999.

Evolutionary Training of SVM for Multiple Category Classification Problems with Self-adaptive Parameters

Ángel Kuri-Morales[1] and Iván Mejía-Guevara[2]

[1] Departamento de Computación, Instituto Tecnológico Autónomo de México,
Río Hondo No. 1,
01000 D. F., México
akuri@itam.mx
[2] Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, IIMAS, Ciudad Universitaria,
04510 D. F., México
imejia@uxmcc2.iimas.unam.mx

Abstract. We describe a methodology to train Support Vector Machines (SVM) where the regularization parameter (C) is determined automatically via an efficient Genetic Algorithm in order to solve multiple category classification problems. We call the kind of SVMs where C is determined automatically from the application of a GA a "Genetic SVM" or GSVM. In order to test the performance of our GSVM, we solved a representative set of problems by applying one-versus-one majority voting and one-versus-all winner-takes-all strategies. In all of these the algorithm displayed very good performance. The relevance of the problem, the algorithm, the experiments and the results obtained are discussed.

1 Introduction

Support Vector Machines have recently received increasing attention from the scientific community due to their solid underlying mathematical foundation. As opposed to more informal (and traditional) alternatives to neural network (NN) development, SVMs rely on well understood mathematical properties which, in effect, allow us to theoretically prove that, for example, perceptron networks (PN) or radial basis function (RBF) ensembles are all encompassed by them. Architectural issues such as the number of hidden layers and the number of neurons in such layers are dispensed with. A number of parameters the user has to heuristically estimate (such as the learning rate in PNs or the number of centers in RBFs) are not present in SVMs. Even though PNs have been extensively studied and applied, it is a striking fact that a large number of parameters (in some variations more than 30) have to be put in "by hand". This has led to the investigation of heuristics which ameliorate the problem. Nonetheless, they remain informal and not necessarily successful in many cases. Similar problems arise with alternative NN approaches.

Given the above, it is all the more interesting to replace the informal NNs by the SVMs which are more general and do not necessitate a heuristic treatment. The architecture of a SVM is shown in figure 1.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 329 – 338, 2006.

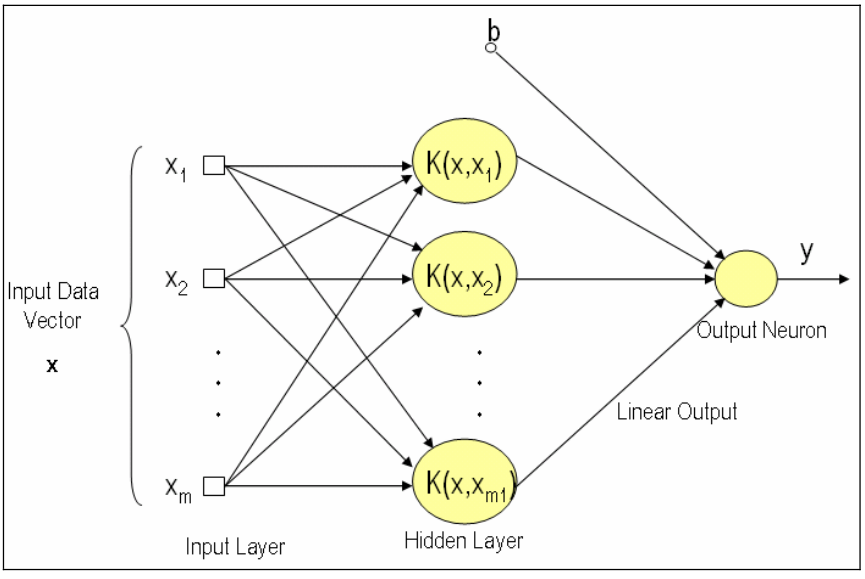


Fig. 1. Architecture of a Support Vector Machine

One key issue in this kind of networks, however, has to do with the so-called "regularization parameter" which, in effect, determines the accuracy of the SVM in terms of possible misclassification of sample elements unknown during the training phase. "C", as of today, has been traditionally determined on a case basis and, although some prior efforts to automate its value do exist [1] there has not been a reliable report of its systematic case-independent automated calculation[1]. In this paper we propose the use of evolutionary computation techniques which help us solve the problem of C's determination; particularly, we focus on multiple category classification problems. In Section 2 we discuss some theoretical issues regarding SVMs, specifically emphasizing the importance of regularization parameter C. In section 3 we discuss how the methodology of Vasconcelos' Genetic Algorithm (VGA) [21] can be used to train this kind of NN and show how to determine automatically the regularization parameter from its application to the dual problem [5]. We also argue that this methodology is appropriate to solve constrained optimization problems, such as these. In section 4 we present three problems we have analyzed in a previous work [20] to show how the GSVM may solve Classification Problems and the resulting level of accuracy. We also discuss the experiments and results. Finally, in Section 5 we offer our conclusions and point to future lines of research.

2 Support Vector Machines

SVM is a supervised neural network that has been used successfully for classification and nonlinear regression problems [2] [3] [4]. In what follows we use the notation "x_i"

[1] There have been some attempts to solve the best determination of the optimal hyperplane without the need for the explicit calculation of "C" [24].

and "w" to denote the independent variable vectors and the weight vectors respectively. A training sample $\tau = \{(x_i, d_i)\}_{i=1}^{N}$ (where x_i is the input pattern for the *i*th example and d_i is the target output) represents two classes in the case of pattern classification.

When attempting pattern classification, the goal is to find a surface that allows the separation of the objects in the sample in two classes: the first class should be on one side of the surface $(d_i = 1)$ and the second class on the other side $(d_i = -1)$. The distance between the nearest points of both classes is called the margin of separation and the optimal surface is found when such margin is maximized.

The form of the surface depends of the linear separability characteristics of τ, i. e., when τ is "linearly separable" the optimal surface corresponds to a hyperplane that is called "Optimal Hyperplane" (OHP) and when τ is "nonlinearly separable", the optimal surface is not a hyperplane in the input space. The introduction of kernel functions is made in order to deal with non-linear decision surfaces. This implies mapping the data to a higher dimensional feature space which allows the construction of an OHP in this space that adequately separates the two classes [5].

The kernel functions are used to map vectors in the input space into vectors in the feature space. These functions must satisfy certain known conditions to be admissible as kernels in a SVM. Specifically they must satisfy Mercer´s condition [5] [6]. Many functions may be used as kernels [7], but the most popular are: a) Polynomial learning machines (PLM), b) Radial-basis functions (RBF) and c) Two-layer perceptrons (LP) [8]. Since the theory allows for any of the above, we used RBF kernels due to their proven simplicity [25].

As mentioned above, we want to find the OHP which maximizes the margin of separation between the two classes that constitute the training set. This gives rise to a constrained optimization problem which has to be solved to get the OHP. A Quadratic Programming (QP) problem is formulated in order to solve it. However, its dual formulation is more adequate, since only the Lagrange Multipliers (LMs) of the QP problem must be found, where these non-zero LMs correspond to support vectors [9]. Thus, the dual form for nonlinearly separable patterns is considered here. Its formulation is as follows:

$$\begin{aligned} &Max\, Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j K(x_i, x_j) \\ &\text{subject to :} \\ &\sum_{i=1}^{N} \alpha_i d_i = 0 \\ &0 \le \alpha_i \le C \quad \text{for } i = 1, 2, ..., N \end{aligned} \tag{1}$$

The solution of this problem is given by α_i, *i=1, 2,..., N* (one α_i for each LM); C is an upper bound for each α_i and $K(x,x_i)$ is a kernel which allows us to construct a decision surface that is nonlinear in the input space but whose image in the feature space is linear.

Regularization Parameter. The upper bound C for the LMs in a nonlinearly separable QP problem is known as "Regularization Parameter" [10]. This parameter controls the trade-off between the complexity of the machine and the level of misclassification allowed. When C is low, a higher proportion of errors is allowed in the solution, while few errors are permissible for high C values.

Automatic determination of C via GA. "C" is traditionally selected by the user. It may be estimated experimentally or analytically [11]. The analytical option relies on the calculation of Vapnik-Chervonenkis (VC) dimension for the problem at hand. VC dimension is, however, extremely difficult to calculate in practice and, in effect, disallows the analytical approach. Therefore, the main goal of this paper is to propose a method to estimate automatically the optimal value of this parameter using a GA without the practical limitations mentioned above. In our approach C's value is in the genome and induces a new constraint. This possibility is typical of the evolutionary approach (and perhaps a few other meta-heuristics) and explains our choice.

3 Genetic Algorithms

GAs are nowadays commonly used to solve complex optimization problems [12]. It is natural to tackle the problem of finding a good value of "C" with one. In what follows we briefly discuss the methodology. In our method, each of the α's in (1) is coded as a binary number which consists of three parts: a sign ("1" stands for "-"; "0" stands for "+"), an integer part and a fractional part. We denote the structure of a "real" (fixed point) valued variable with SI.F (for example S4.20) which stands for "one bit for the sign, "I" bits for the integer part and "F" bits for the fractional part" (in the example we indicate 4 bits for the integers and 20 bits for the decimals). This fixed point representation is free from the Hamming cliffs which present themselves when using the floating point representation alternative. Therefore, a set of N α's will be represented by a continuous binary string of length N x $(I+F+1)$ bits. Additionally, we include the hypothetical value of "C" as part of the chromosome in a similar format. Therefore, the length of the chromosome turns out to be $(N+1)(I+F+1)$ bits. With this representation, an x variable's value range is, approximately, $-2^I \leq x \leq +2^I$.

The optimization process is complex enough so that a traditional simple genetic algorithm (SGA) will frequently exhibit a poor behavior. Therefore, we replace the SGA by another variation of a GA which is called Vasconcelos' GA (VGA). In VGA the traditional proportional selection is replaced by a deterministic scheme in which the elements of the population are sorted according to their fitness and the best individual is coupled with the worst individual. The resulting crossover will guarantee that the genetic diversity is preserved, as was shown in [26]. For this scheme to lead to an evolutionary improvement, however, we need to establish a complementary strategy whereupon the best individuals are retained. From the deterministic pairing and crossover P new individuals will result. That is, a population of size P will yield a set of P new individuals. Then, out of a total of 2P individuals (P original ones and P new ones) the overall best P are retained while the remaining (worst) P are discarded. This process has shown to yield very good results (relative to an SGA).

3.1 Training a SVM Using GAs

Several commercial optimization libraries can be used to solve the quadratic programming problem. However, these libraries are of limited use. The memory requirements of the QP problem grow with the square of the size of the training sample [13]. For this reason, in "real life" applications, the QP problem cannot be easily solved using commercial optimization libraries. Although some optimization techniques can be directly applied to QP problems, many of them require the kernel matrix to be stored in memory, implying that the space complexity grows as the square of the sample size. For large sized problems, these approaches are impractical and must, therefore, be used in conjunction with other techniques [14]. In this paper, we use GAs to tackle the QP problem.

GAs as an optimization tool. The application of GAs to SVMs differs substantially from previous approaches to train NNs because the dual QP problem presented above is used to find the support vectors directly. In previous works [22] [23] the support vectors have been determined from the application of Lagrange Multipliers, which neatly adjust to this problem and satisfy the Karush-Kuhn-Tucker conditions, but are not applicable to search for "C" [11]. In fact, GAs are used here to solve the constrained QP. One advantage of using GAs for this kind of problems is that restrictions are not imposed in the form of the objective function: neither the objective function nor the constraints of the problem must be (for example) derivable in order to solve de problem. In our approach, each individual represents a LM (α_i, i=1,...,N), where N is the number of points in the training set for the dual SVM problem. Every variable is to be expressed in fixed point format with one sign bit (0→+; 1→-), 4 integer bits and 20 decimal bits as shown in figure 2.

α_i		
Sign	**Integer**	**Decimal**
1 bit	4 bits	20 bits

Fig. 2. Fixed point representation

3.2 Relative Optimality of VGA

Although GAs were originally designed to solve unconstrained optimization problems, they can be adapted to tackle the constrained cases [15] as will be shown. On the other hand, out of the large supply of possible GA's variations we have selected the so-called VGA, for the reasons mentioned above.

The first step is the selection of the population's size. In this work we considered a population of size P = 100 for all of the problems; the initial population was randomly generated; weighted binary fixed point representation, as previously discussed, was used. Each individual represents a Lagrange multiplier (α_i, $i=1,...,N$), where N is the number of points in the training set for the dual SVM problem.

With this representation the genome's size is (N+1)x25, where the first 25N bits correspond to the number of objects in the training sample and the last 25 bits correspond to the value of C. Once the initial population is generated, VGA is used with

P_m=0.05 (P_m = probability of mutation) and P_c=0.9 (P_c = probability of crossover). The evaluation function is constructed following the methodology of SVMs but we modify it by transforming the constrained original problem to a non-constrained one. To do this, we have chosen the penalty function (*F(x)*) [16]:

$$F(x) = \begin{cases} \left[Z - \sum_{i=1}^{s} \frac{Z}{t} \right] - f(x) & s \neq t \\ 0 & otherwise \end{cases} \tag{2}$$

where *Z* is a large constant [$O(10^9)$], *t* is the number of constraints, *s* is the number of these which have been satisfied and *f(x)* corresponds to the fitness at point *x*. The GA operation was terminated after 100 generations.

4 Training SVMs with GAs for Multiple Category Classification Problems: Experiments and Results

We have applied the methodology to the problem of determining whether an object from a sample belongs to one of two classes. This may be extended to N classes [17]. SVMs have traditionally been designed to deal with binary classification, but most real world problems have more than two classes.

The classification problems which we used to test our method consist of a set of data which correspond to three classes. SVMs, powerful as they are, do not lend themselves to an easy treatment of more than two classes. There are two possible approaches to the generalization of logistic (two set) classification with SVMs. In the first one, one could attempt to tackle the multi-class by transforming the classification problem into a regression one. That is, one may assume that the data belongs to a class if the SVM outputs the value of a function which maps such data to a real value within pre-established values and each one of these values may be set arbitrarily. For instance, one could assign the value "-1" to the elements in set A, the value "0" to the elements in set B, the value "+1" to those elements in set C and so on. The other (and most common alternative) is to treat the classification problem as a logistic one and then combine partial classifications to attempt a multi-class assignment. This is the strategy we followed and which is reported here.

For variety and comparison a one-versus-one classifier and a one-versus-all classifier [18] were used. In a one-versus-one classifier, a SVM model is built for each pair of classes. This results in *p(p-1)/2* SVM classifiers (*p* is the number of classes in a specific problem). In a one-versus-all classifier, *p* classifiers are used. The ratio between the number of classifiers in a one-versus-one classifier and a one-versus-all classifier is *(p-1)/2*, which is significant when *p* is large. On the other hand, all *N* objects are used in each classifier in a one-versus-all classifier.

A set of classification problems is presented here in order to illustrate the classification efficiency of the method. A similar set was discussed in [20] where, however, the characteristics of the chromosome were somewhat different. A pre-specified percentage of the elements of the data set is randomly selected yielding the test set. The remaining elements correspond to the training set. The SVM is trained with the training set; its generalization properties are enhanced by selecting the support vectors

which minimize the error for the test set. The problems are: 1) Lung Cancer Database: we got 180 objects from natural spline interpolation out of a very small set [19]: 150 were used for training and 30 for testing; 2) Wine Recognition Database: it consists of three types of wines with 13 real valued attributes and 178 instances: 59 for class 1, 71 for class 2 and 48 for class 3; and 3) Iris Plant Database: which consists of 3 classes of 50 instances each and four attributes. The details may be found in [20].

In this paper we use the majority voting scheme to measure the accuracy of the one-versus-one method where, for a new example x_i, class l wins a new vote if the classifier l_m (means "class l vs class m") says x_i is in class l. Otherwise, the vote for class m is increased by one. After each one of the $p(p-1)/2$ binary classifiers casts its vote, the majority voting scheme assigns x_i to the class with the largest number of votes. In the case of one-versus-all method, the accuracy is measured using the winner-takes-all strategy where x_i (for a new element x_i) is assigned to the class with the largest output [18].

It is important to mention that, after vast experimentation with several plausible values of parameter σ we settled with a Radial Basis Function Kernel with $\sigma = 2$. This value was used in every experiment (RB2 in tables 1, 2 and 3). As mentioned in the initial paragraphs, SVMs encompass the Radial Basis Function NNs. These functions share some nice mathematical properties. However, it is natural to enquire as to the appropriateness of their use in this case. Since alternative kernels are also possible it is not evident which of the many (in fact, infinite) alternatives is best. This poses no problem for SVMs since the underlying theory is not dependent on the kernels. Considerations leading to the best choice are of a more practical nature. Our choice reflects ease of programming and numerical precision of the results.

4.1 Results

Lung Cancer. In this problem, 3 classes are considered, hence, one-versus-one and one-versus-all classifiers are used in order to test the methodology proposed here. The results of the application of these methodologies are shown in Table 1 and Table 2, respectively. We obtained 99.3% accuracy for training data, 96.7% for test data when using a majority voting strategy; on the other hand, the SVM achieved 92.0% accuracy for training data and 93.3% for test data when using a winner-takes-all strategy.

Wine Recognition. One-versus-one and one-versus-all classifiers were used and three classes considered in this problem. Hence, this results in 3 SVM classifiers for each alternative. The corresponding results are shown in Table 1 and Table 2, respectively. The accuracy of both classifiers is very similar. The one-versus-one classifier with majority voting strategy had an accuracy of 96.0% for training data and 92.9% for test data. The one-versus-all classifier with the winner-takes-all strategy had an accuracy of 94.7% for training data and 92.9% for test data.

Iris Plant. One-versus-one and one-versus-all classifiers were used for this recognition problem. The accuracy for both training and test sample was 88.0%, applying the majority voting strategy (Table 1) and 97.6% for training sample, 100% for testing sample, applying winner-takes-all strategy (Table 2).

Table 1. Results of one-versus-one method for Lung Cancer, Wine Recognition and Iris Plant problems

one-vs-one				
Problems	**Training Accuracy**	**Test Accuracy**	**Average C**	**Kernel**
Lung Cancer	99.3%	96.7%	4.17	RB2
Wines	96.0%	92.9%	13.67	RB2
Iris	88.0%	88.0%	12.03	RB2

Table 2. Results of one-versus-all method for Lung Cancer, Wine Recognition and Iris Plant problems

one-vs-all				
Problems	**Training Accuracy**	**Test Accuracy**	**Average C**	**Kernel**
Lung Cancer	92.0%	93.3%	12.00	RB2
Wines	94.7%	92.9%	11.00	RB2
Iris	97.6%	100.0%	12.01	RB2

5 Conclusions

A GSVM classifier is presented in this paper. The application of this algorithm to a set of test problems resulted in very good performance. The application of a VGA allows us to tackle an extremely complex constrained optimization problem in a very simple and straightforward way (if judged from the traditional point of view). Consider that every one of the data vectors determines a constraint. For example, in a typical problem the number of constraints is larger than 150. The VGA has to determine the set of feasible values out of a very large set. However, the most important issue is that the value of the regularization parameter was automatically found through the algorithm rather than put in by hand. The reported work seems to indicate that the VGA (along with proper constraint handling) is an efficient way to optimize C by including it in the genome. In the past, the difficulty of properly determining the value of C was usually interpreted, simply put, as changing one typical problem in NNs (the determination of the most adequate architecture) into another, perhaps more difficult, one (the best determination of the regularization parameter). By automatically determining the value of C (as we have shown) the theoretical advantages of SVMs may be fully exploited and the negative issue mentioned above may be by-passed.

Acknowledgement

We want to thank UCI Machine Learning Repository for the use of Lung Cancer Database, Wine Recognition Database and Iris Plant Database.

References

1. Jordaan, E. M. & Smits, G. F.: Estimation of the regularization parameter for support vector regression. Proc. of World Conference on Computational Intelligence I. Honolulu, Hawaii. (2002) 2785-2791.
2. Schmitdt, M., Grish, H.: Speaker identification via support vector classifiers. Proc. of International Conference on Acoustics, Speech and Signal Processing, (1996) 105-108.
3. Drucker, H., Wu, D., Vapnik, V.: Support vector machine for spam categorization. Trans. on Neural Networks, Vol. 10. IEEE, (1999) 1048-1054.
4. Vapnik, V., Golowich, S., Smola A.: Support vector method for function approximation, regression, estimation and signal processing. Adv. Neural Inform. Process. Syst., Vol. 9. (1996) 281-287.
5. Haykin, S.: Neural Networks. A comprehensive foundation. 2nd ed. Prentice Hall, New Jersey (1999).
6. Mercer, J.: Functions of positive and negative type, and their connection with the theory of integral equations. Transactions of the London Philosophical Society (A), Vol. 209. (1909) 415-446.
7. Shawe-Taylor, J., Cristianini, N,: Kernel Methods for Pattern Analysis. Cambridge University Press, UK (2004).
8. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, UK (2000).
9. Burges, C. J. C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, Vol. 2. (1998) 121-167.
10. Haykin, S., op. cit., pp. 318-350.
11. Haykin, S., op. cit., pp. 326-329.
12. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge, Massachusetts (1996).
13. Burges, C. J. C.: op. cit. 121-167.
14. Cristianini, N., op. cit., pp. 93-124.
15. Kuri, A., Gutiérrez, J.: Penalty Functions Methods for Constrained Optimisation with Genetic Algorithms. A Statistical Analysis. Lecture Notes in Artificial Intelligence, Vol. 2313. Springer-Verlag, (2002) 108-117.
16. Kuri, A., Gutiérrez, J., op. cit., pp.109-111.
17. Fung, G., Mangasarian, O. L.: Multicategory proximal support vector machine classifiers. Data Mining Institute Technical Report, (2001) 01-06.
18. Allwein, E. L., Shapire, R. E., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research, Vol. 1, (2000) 113-141.
19. Bojanov, B., Hakopian, H., Sahakian, B.: Spline Functions and Multivariate Interpolations. Springer-Verlag, (1993).
20. Kuri, A., Mejía, I.: Evolutionary Training of SVM for Classification Problems with Self-Adaptive Parameters. Advances in Artificial Intelligence Theory, (Gelbukh, A., Monroy, R., Eds.), IPN, (2005), pp. 207-216.
21. Kuri Morales, A.: A Methodology for the Statistical Characterization of Genetic Algorithms. Proceedings of the Mexican International Congress on Artificial Intelligence, Springer-Verlag, (2002) 79-88.
22. Vapnik, V.: Estimation of Dependences Based on Empirical Data. Springer-Verlag, (1982).

23. Osuna, E., Freund, R., Girosi, F.: Improved training algorithm for support vector machines. Proc. IEEE Neural Networks in Signal Processing '97, (1997).
24. Mangasarian, O.L., Support Vector Machine Classification via Parameterless Robust Linear Programming, Data Mining Institute Technical Report 03-01, (2003).
25. Haykin, S., op. cit., 293-294.
26. Kuri Morales, A.: A Methodology for the Statistical Characterization of Genetic Algorithms, Lecture Notes in Artificial Intelligence: No. 2313, (Eds) Coello, C., Albornoz, A., Sucar, E., Cairó, O., Springer-Verlag, ISSN 0302-9743, ISNN 3-540-43475-5, (2002) 79-88.

Time-Space Ensemble Strategies for Automatic Music Genre Classification

Carlos N. Silla Jr., Celso A.A. Kaestner, and Alessandro L. Koerich

Pontifical Catholic University of Paraná (PUCPR)
Postgraduate Programme in Computer Science (PPGIA)
Rua Imaculada Conceição 1155, 80215-901 Curitiba, Brazil
{silla, kaestner, alekoe}@ppgia.pucpr.br

Abstract. In this paper we propose a novel time–space ensemble–based approach for the task of automatic music genre classification. Ensemble strategies employ several classifiers to different views of the problem–space, and combination rules in order to produce the final classification decision. In our approach we employ audio signal segmentation in time intervals and also problem space decomposition. Initially the music signal is split in time segments; features are extracted from these music signal segments and the one against all (OAA) and round robin (RR) strategies, which implement a space decomposition by using several binary classifiers, are applied. Finally, the outputs of the set of classifiers are combined to produce the final result. We test our proposition in a music database of 1.200 music samples from four different music genres. Experimental results show that time segment decomposition is more important than the space decomposition produced by the OAA and RR strategies, although they produce better results relative to the use of single classifiers and feature vectors.

1 Introduction

The large amount of multimedia information on the web surface nowadays makes it necessary to build a new class of automatic tools, capable of dealing with information from very different media [1]. In this context one of the most important tasks is the automatic content-based music genre classification. The music genre is one of the most important aspects to describe music, and it is mainly used to organize large collections of digital music [2].

From a pattern recognition perspective, music genre classification poses an interesting research problem, since music is a complex time-variant sinal. Another interesting aspect is that genre classification is naturally a multi-class problem. In order to deal with multi-class problems there are two basic possibilities: the first one is to use learning and classification techniques that can naturally handle multi-class problems – producing complex decision surfaces – like decision trees, k nearest neighbors (k-NN), neural networks, etc.; the second option is to use a problem space decomposition strategy to break a multi-class problem into a series of binary problems that can be tackled using a set of binary classifiers – which produce simple decision surfaces – like support vector machines (SVM).

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 339–348, 2006.

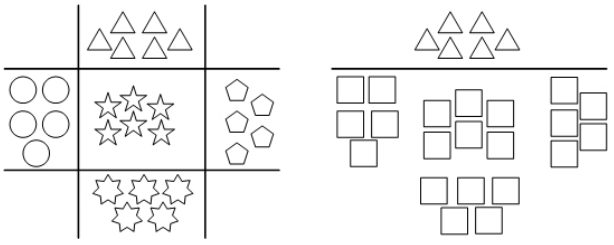


Fig. 1. An example illustrating the problem space decomposition strategy

In several pattern recognition areas [3] [4] the so called "ensemble approach" has been used with success. This approach consists of applying to the problem not a single classifier, but a collection of them, each one specialized in a specific view of the problem. In this way, each classifier is trained on different distributions, and the outputs of predictors are combined by a dynamic classifier combination model. This procedure may be viewed as either a version of mixture of experts [5] applied to classification, or a variant of the boosting algorithm [6]. A possible explanation for the success of the ensemble approach is that classifiers applied to partial views of the problem space produce simpler decision surfaces, and therefore, better classification results. Figure 1 shows an example of such an approach.

Fürnkranz [7] suggests that the problem space decomposition strategy can be used as an ensemble technique for any classifier (binary or not). A common problem space decomposition strategy is the one against all (OAA), where a classifier is created to recognize the set of patterns that belongs to one specific class. A second problem space decomposition strategy is the round robin (RR) [8] (a.k.a. pairwise comparison [7]) where a set of classifiers is created for every possible combination of two classes. A third possible approach is the random subspace method (RSM) [9], where classifiers are applied to a set of random selected projections of the problem space. In all cases the individual classification results are combined to produce the final classification.

The idea of developing an ensemble strategy based on problem space decomposition for the task of music genre recognition was introduced by Grimaldi et al. [10,11]. They evaluate the performance of different ensemble methods (OAA, RR, RSM). The experiments were performed using features based on the discrete wavelet packet transform (DWPT), which where extracted from the whole music signal. A major limitation in their work is that the efficiency of the ensemble methods were not evaluated alone, since the presented results also include feature selection. Li et al. [8] present an analysis of different classifiers applied to the music genre recognition task. In this work they employ the OAA and the RR problem space decomposition techniques using a set of SVM classifiers. The experimental results show that the OAA strategy achieves slightly better results than the RR strategy. One limitation of this work is that the decomposition strategies were evaluated only with SVM classifiers.

Costa et al. [12] exploit a different way of using the ensemble approach for music classification. In their work each classifier of the ensemble was trained and applied to different time intervals – or segments – of the music, taking into account the temporal nature of the music signal. They separate three segments from the beginning, middle and end of the music signal, in order to produce a set of feature vectors and obtain the corresponding classifications. Music classification problems usually employ the beginning or the whole music signal in order to deal with classification [1], [11]. Costa et al. justify their proposition because a music signal can strongly vary in the time dimension – for example a rock music can start as a classical music, so it seems to be a good idea to treat different segments and combine the obtained classification results. In their work the k-NN classifier and multilayer perceptron (MLP) neural networks classifiers were used in each segment.

In this paper we apply the ensemble approach for automatic music genre classification in a new way. We consider the time dimension of the music signal and also space decomposition by using OAA and RR strategies for sets of binary classifiers, attempting to exploit the advantages offered by both strategies. The performance of the proposed approach is compared with other ensemble methods. We also employ a broader range of classifiers than the ones used in the previous works. This paper is organized as follows: Section 2 presents the foundations of the ensemble methods. Section 3 describes the target music genre classification task. Section 4 presents the experiments carried out and an analysis of the achieved results. Finally the last section presents our conclusions and concluding remarks.

2 Ensemble Methods

An "ensemble approach" to a pattern recognition problem consists of a decomposition of the original problem space by using a collection of classifiers, each of them dedicated to a specific view of this space. In a first step each classifier – usually a binary one – is trained and applied to its view, producing an individual classification; in a second step these classification results are dynamically combined to produce the final classification.

The main motivation to apply problem-space decomposition using methods like OAA and RR is that multi-class classification is intrinsically harder than binary classification, because the classification algorithm has to construct a high number of separation boundaries, whereas binary classifiers have to determine only one appropriate decision function [13].

2.1 One Against All (OAA) Strategy

Given a n-class pattern recognition problem, the OAA strategy consists of creating a set of n binary classifiers, one for each class. Each classifier is trained through re-labeling of the same training dataset, in order to distinguishing between one single class and its complement in the problem space. For instance,

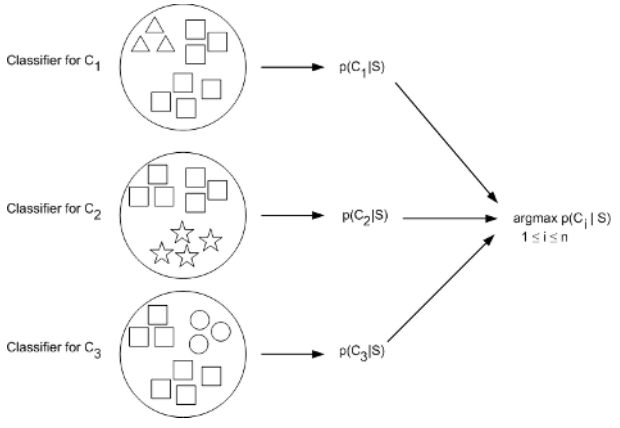


Fig. 2. Illustration of the OAA strategy for a three class problem

the classifier for class C_i is trained using the elements of C_i as positive examples and the remaining of the data-set as negative examples, producing a specialized classifier for class C_i. For an unseen example represented by a feature vector s, given the n individual classifications, and considering that each individual classifier assigns to s a probability p (or a confidence score) that is directly related to the conformity of this example with its class, the final class $\hat{C}$ assigned to s is given, as usual, by:

$$\hat{C} = C_k | k = \arg\max_{1 \leq i \leq n} p(C_i|s) \tag{1}$$

where $p(C_i|s)$ is the a posteriori probability of class C_i given a feature vector s and $\hat{C}$ is the winner class, that is, the one which provides the highest a posteriori probability. Figure 2 illustrates this approach.

2.2 Round Robin Strategy

Fürnkranz [7] presents the RR problem space decomposition as an ensemble strategy, in order to allow binary classifiers to deal with multi-class problems. The RR method converts a n-class problem into a series of binary problems, by creating a set of $k = n(n-1)/2$ classifiers, one for each pair of classes. Unseen samples are classified by presenting them to the set of k binary classifiers. In this case when an unseen example e is presented to each one of the k binary classifiers, a class is directly assigned to e. The k assigned classes are finally combined into the final result through majority voting.

Contrary to the OAA ensemble strategy, in this case when a binary classifier is constructed, let's say for classes C_i and C_j, only the examples of these two classes are used, and the rest of the dataset is ignored. According to Fürnkranz [7], this leads to an easier decision about the boundary between the two classes. Figure 3 illustrates this approach.

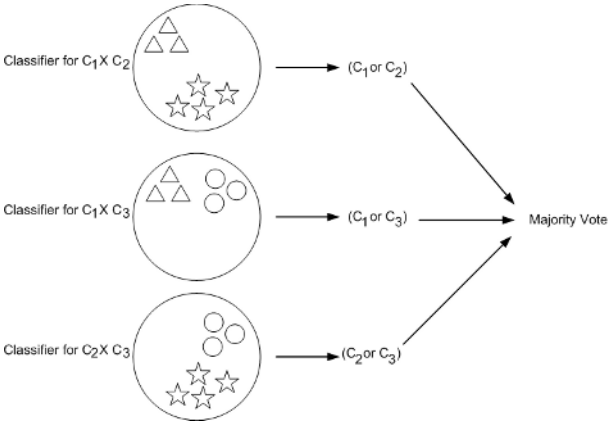


Fig. 3. Illustration of the RR strategy for a three class problem

2.3 Segment–Based Ensemble Strategy

A different ensemble strategy was proposed by Costa et al. [12] for the task of automatic music genre classification. The proposed approach can be easily extended to other time-variant signals or to time-dependent classification tasks. The music signal m is split into time intervals or segments, and features are extracted separately from each segment. The same features are extracted from each segment and three classifiers are trained.

However, music is a time-variant signal, and the decomposition is made according to the time dimension, producing different views of the same object. In this case, as already noted by Kittler et al. [3], it is possible to use alternative ensemble methods. When a new unseen music m' is presented, the corresponding temporal segment are extracted, producing three different views of m'. The specific classifier of each segment is then applied, and the final classification decision (in this case, the music genre) is carried out through the majority voting principle.

2.4 A Time–Space Ensemble Approach

It is possible to combine both ensemble techniques described previously in order to perform a new ensemble-based approach to music genre classification. In the first (time decomposition) step the music signal is segmented according to a set of time intervals. Features are then extracted from these segments and used in a second ensemble decomposition (space decomposition) using OAA and RR strategies and a set of binary classifiers. Finally, a compositional rule is employed, taking into account the individual classification decisions (for both time and space decompositions) to provide the final class label. Figure 4 illustrates the proposed approach.

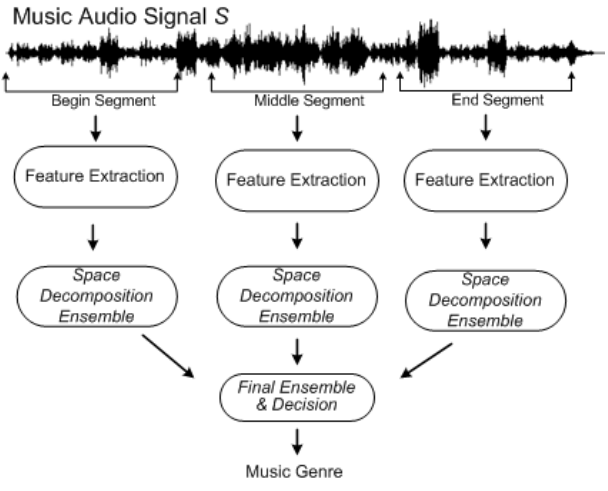


Fig. 4. The main components of the time–space ensemble–based approach

3 Music Genre Classification

The problem of music genre classification can be formally defined as the task of assigning a boolean value T ($true$) or F ($false$) to each pair $\langle m, g\rangle \in \{\mathcal{M} \times \mathcal{G}\}$, where $\mathcal{M}$ is a domain of music (signals) and $\mathcal{G}$ is a predefined set of music genres. A value T assigned to $< m, g >$ indicates that m belongs to the genre g, while F indicates that m does not belong to g. In a classification process we try to approximate an unknown *target function* $\Phi^{real} : \mathcal{M} \times \mathcal{G} \rightarrow \{T, F\}$ by means of a function $\Phi : \mathcal{M} \times \mathcal{G} \rightarrow \{T, F\}$ called the classifier (a.k.a. rule, or model) such as Φ^{real} and Φ "coincide as much as possible" [14].

In our case, the final classifier Φ is not obtained directly from the whole music signal, but from the music segments and a set of classifiers, according to the OOA or the RR strategy. In each segment and binary space decomposition we employ conventional algorithms.

We split each music signal into three 30 seconds segments – which is often used in the literature [1], [8] – from the beginning, middle and end of the music signal. If a music signal is composed by f frames, we consider the beginning of the music the frames from 0 to 1.153 (corresponding in a MP3 file to 30 seconds), the middle segment as the frames from $(\frac{f}{3} + 500)$ to $(\frac{f}{3} + 1.453)$ and the final segment as the frames $(f - 1.453)$ to $(f - 300)$. During the segmentation stage each segment from the audio signal is extracted after the conversion of the MP3 File to 16 bits mono Wav.

The feature extraction from the music segments is carried out through the Marsyas framework [15], which implements the original feature set proposed by Tzanetakis & Cook [1]. The features used in this work can be divided into three groups: timbral texture, beat related and pitch related. The features based on the timbral texture account for the mean and variance of the spectral centroid,

rolloff, flux, the time zero domain crossings, the first 5 MFCC's and low energy. Features that are beat related include the relative amplitudes and the beats per minute. Pitch related features include the maximum periods of the pitch peak obtained from the pitch histograms. All this features are concatenated to form a 30-dimensional feature vector. More details about these features can be found in [1].

We use the following machine learning algorithms as component classifiers for the ensemble methods: Decision Trees (DT), k-NN, Naïve Bayes (NB), Support Vector Machines (SVM) with pairwise classification and an MLP neural network classifier trained with the backpropagation momentum algorithm [16]. The employed classification framework is based on the Weka Datamining Tool [17].

The final classification result is obtained from these partial classifications, by means of a combination rule. In our case the partial results are combined by using the majority vote rule. It is important to notice that we have not mixed different component classifiers into the ensembles. The ensembles are made up by homogeneous classifiers.

4 Experimental Results

The main objective of our experiments is to evaluate the performance of the different ensemble methods for the task of automatic music genre classification, and if it is advantageous or not to combine time segmentation and problem space decomposition into a unique ensemble approach to deal with this classification problem.

A database containing music samples from four different Latin genres was available for the experiments. We have selected 300 samples from each genre (Tango, Salsa, Forró, and Axé) and split them into training (50%), validation (20%) and test (30%) sets according to a classical holdout procedure. Due to time constraints cross-validation was not employed. The music samples have been selected randomly from a large database without reposition to avoid bias in the experiments.

In order to have a baseline (BL) we have also used the classifiers in a conventional manner – so, with no space decomposition – to each music segment. Also it is important to note that since the baseline SVM classifier uses the RR decomposing strategy, its results in both columns are the same.

Table 1 presents the accuracy results achieved by the different classifiers on individual segments. It seems that for the beginning segment the best classification accuracy is often achieved by the RR ensemble strategy. This is not true only for the DT classifier which presents the best result without using any ensemble technique and the NB classifier which presents the best result using the OAA ensemble. For the 1-NN the result is the same regardless of the strategy employed.

For the middle segment, the RR ensemble often presents the best classification accuracy. Similar to the beginning segment, the DT holds better results using only a BL classifier. For the 1-NN and the NB classifier, regardless of the strategy

Table 1. Accuracy (%) using different strategies on individual music segments

	Begin			Middle			End		
Classifier	BL	OAA	RR	BL	OAA	RR	BL	OAA	RR
DT	**67.22**	63.05	64.72	**68.61**	65.27	66.94	50.00	58.88	**67.22**
1-NN	68.33	68.33	68.33	73.61	73.61	73.61	68.05	68.05	68.05
3-NN	67.22	67.22	**68.05**	75.83	75.83	**76.66**	71.66	71.66	**73.05**
5-NN	68.33	68.33	**70.27**	76.38	76.38	**76.66**	**70.27**	**70.27**	69.72
7-NN	70.83	70.83	**71.94**	74.44	74.44	**75.83**	**73.05**	**73.05**	70.27
MLP	76.94	80.55	**83.33**	80.83	**85.27**	77.77	66.11	**68.33**	59.72
NB	69.16	**70.27**	69.16	76.94	76.94	76.94	61.94	**62.50**	61.66
SVM	**81.11**	66.38	**81.11**	**86.66**	74.44	**86.66**	**70.00**	60.27	**70.00**

adopted, the results are similar. Also for the MLP classifier, the best accuracy was achieved using the OAA ensemble.

For the segment that represents the end part of the music, the OAA ensemble often presents the best results. This is true for the MLP and the NB classifier and also for the 1-NN, 5-NN and 7-NN classifier. However, the results achieved using the OAA ensemble of k-NN classifiers are similar to the results achieved by the BL approach. The RR ensemble outperformed the OAA only for the DT and 3-NN classifier.

In summary, for the initial segment the best result was achieved using the MLP with RR (83.33%); for the middle segment the best result was achieved using the SVM with RR (86.66%); for the end segment the best results were achieved using the 3-NN with RR, and 7-NN with OAA (73.05%).

It is also possible to combine the decisions of the ensemble of classifiers specialized in each segment of the music and build a time-space ensemble approach based on majority voting which takes into account the final class label provided by the ensemble performed on each music segment. The accuracy results achieved by this approach are presented in Table 2.

Table 2 shows that the ensemble of the DT individual results is the same as for the for the RR ensemble. For the k-NN classifiers using only the individual segments often achieves better results than using the OAA or RR ensembles. This is not true only for the 5-NN, where the RR time-space ensemble performs slightly better. For the MLP and the NB classifier the time-space OAA ensemble strategy provides better results than using the BL classifiers or the RR ensemble. For the SVM classifier the results achieved are the same for the BL and the RR ensemble, because as mentioned before, the decomposition strategy used for handling multi-class problems in the SVM was the RR.

When evaluating the results achieved by the ensembles and comparing them with the results achieved by the classifiers on the individual segments, we see that the majority vote ensemble provides better accuracy in any case only for the DT and k-NN. For the MLP classifier, the ensemble of BL classifiers provides better results than any of the individual segment/classifier. This does not happen for the time-space ensembles which present better results using only one of the individual segments. The majority vote ensemble does not improve

Table 2. Accuracy (%) using the majority vote rule for segment ensemble

	3 Segment Majority Vote		
Classifier	BL	OAA	RR
DT	**75.27**	73.61	**75.27**
1-NN	**78.33**	**78.33**	78.05
3-NN	**81.38**	80.55	81.11
5-NN	81.38	81.11	**81.94**
7-NN	**82.50**	81.94	80.83
MLP	81.38	**83.33**	79.72
NB	72.50	**76.94**	72.22
SVM	**86.11**	77.50	**86.11**

the performance of the NB classifier as the best accuracy is achieved with the middle segments. For the SVM classifier the accuracy using the middle segment outperforms the performance of the time-space ensembles.

5 Concluding Remarks

In this work we propose a new approach to the task of automatic music genre classification based on time-space ensemble decomposition. Time decomposition is achieved by breaking up the music signal in several temporal segments. The proposed approach uses three feature vectors extracted from the beginning, middle and end parts of the music. This procedure tries to assure that the most important temporal patterns of the music signal to be considered in the classification. Space decomposition is obtained by applying sets of binary classifiers to a naturally multi-class problem. This procedure tends to produce simpler separation surfaces in specific views of the problem space. We employ two different space-decomposition ensemble strategies, namely One-Against-All (OAA) and Round Robin (RR). Final results are obtained by simple composition rules. We use the classical DT, Naïve Bayes, k-NN and SVM as classification algorithms.

The experiments were performed using the classifiers and a large dataset composed by 1.200 music samples of different Latin music genres, namely Tango, Salsa, Forró and Axé. The achieved results show that the accuracy in music genre classification achieved by the RR decomposition provides better results than the OAA ensemble and baseline classifiers when considering the individual segments. Contrary to our expectative, the complete ensemble approach – using time and space decompositions – does not present superior results in comparison with the OAA and RR ensemble strategies on individual segments.

One solution that might improve the performance of the time-space ensemble approach in this scenario would be to base the final decision on the vote of all space component classifiers, instead of using only the final label provided in each segment. For example, in the case of the RR approach it would be possible to make the final decision based on the majority vote of all the trained classifiers instead of only the output produced by each segment. Also it would be possible

to use more robust rules instead of only the majority voting. Both aspects are subject of our current research.

References

1. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing **10** (2002) 293–302
2. Aucouturier, J.J., Pachet, F.: Representing musical genre: A state of the art. Journal of New Music Research **32** (2003) 83–93
3. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transaction on Pattern Analysis and Machine Intelligence **20** (1998) 226–239
4. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research **5** (2004) 101–141
5. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptative mixtures of local experts. Neural Computation **3** (1991) 79–87
6. Schapire, R.: The strength of weak learnability. Machine Learning **5** (1990) 197–227
7. Frnkranz, J.: Pairwise classification as an ensemble technique. In: Proceedings of the 13th European Conference on Machine Learning, Helsinki, Finland, Springer-Verlag (2002) 97–110
8. Li, T., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. In: Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, Canada (2003) 282–289
9. Ho, T.K.: Nearest neighbors in random subspaces. In: Proc. of the 2nd Int'l Workshop on Statistical Techniques in Pattern Recognition. (1998) 640–648
10. Grimaldi, M., Cunningham, P., Kokaram, A.: An evaluation of alternative feature selection strategies and ensemble techniques for classifying music. In: Workshop on Multimedia Discovery and Mining at ECML/PKDD-2003. (2003)
11. Grimaldi, M., Cunningham, P., Kokaram, A.: A wavelet packet representation of audio signals for music genre classification using different ensemble and feature selection techniques. In: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, ACM Press (2003) 102–108
12. Costa, C.H.L., Valle Jr, J.D., Koerich, A.L.: Automatic classification of audio data. In: IEEE International Conference on Systems, Man, and Cybernetics. (2004) 562–567
13. Dietterich, T.G.: Ensemble methods in machine learning. In: Proc. 1st International Workshop on Multiple Classifier System. Number 1857 in Lecture Notes in Computer Science, Springer (2000) 1–15
14. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
15. Tzanetakis, G., Cook, P.: Marsyas: A framework for audio analysis. Organized Sound **4** (1999) 169–175
16. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
17. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2nd edn. Morgan Kaufmann, San Francisco (2005)

Predictive and Descriptive Approaches to Learning Game Rules from Vision Data

Paulo Santos[1], Simon Colton[2], and Derek Magee[3]

[1] Department of Electrical Engineering
Centro Universitário da FEI, São Paulo, Brazil
psantos@fei.edu.br
[2] Department of Computing, Imperial College, London, UK
sgc@doc.ic.ac.uk
[3] School of Computing, Leeds University, Leeds, UK
drm@comp.leeds.ac.uk

Abstract. Systems able to learn from visual observations have a great deal of potential for autonomous robotics, scientific discovery, and many other fields as the necessity to generalise from visual observation (from a quotidian scene or from the results of a scientific enquiry) is inherent in various domains. We describe an application to learning rules of a dice game using data from a vision system observing the game being played. In this paper, we experimented with two broad approaches: (i) a predictive learning approach with the Progol system, where explicit concept learning problems are posed and solved, and (ii) a descriptive learning approach with the HR system, where a general theory is formed with no specific problem solving task in mind and rules are extracted from the theory.

1 Introduction

For full autonomy, agents will require higher level abilities to reason about low level data collected from the environment, so that they can adapt to the environment and make discoveries which are necessary for the completion of tasks. Towards this goal, [12] proposes a cognitive vision system capable of learning protocols from the visual information of dynamic scenes using an inductive logic programming system (ILP). In that work the predictive ILP learning system Progol [10] was used to induce theories from the visual input system. In a *predictive* learning system a particular categorisation problem, or a set of such problems, are specified, and the learning system derives a theory which performs well when the derived rules in the theory are used for predicting the category of unseen examples. ILP systems which perform well at these tasks include the Progol [10] and FOIL [13] systems. An alternative mode within which ILP can be applied is known as *descriptive* learning. In such approaches, there is no particular problem to solve, other than that of discovering interesting concepts relating to the data provided and interesting facts which relate these concepts. Descriptive learning programs, such as the HR system [1], the CLAUDIEN [4] and WARMR [5] systems, produce theories containing examples, concepts which categorise the examples, conjectures which relate the concepts and explanations which illustrate the truth or falsity of the conjectures. The purpose of the experimentation and the reporting of it here is

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 349–359, 2006.

primarily to compare and contrast descriptive and predictive approaches to learning rules from vision data. More specifically, the contribution of this paper is a comparison between Progol and HR on the task of learning protocols from visual observations reported in [12] whereas, in this earlier work, only Progol was employed. While qualitative comparisons of these systems are given in [1] and [2], this is the first quantitative comparison between them.

It is worth pointing out that, while there have been many approaches to learning from visual data (as overviewed in [12]), in this work we are interested in, first, to have access to the sound and completeness of the knowledge learned (cf. [12]) and, second, to further use the learned rules for higher level spatial and temporal reasoning, as experimented in [8]. Thus, we chose to employ an ILP system for the learning tasks.

The paper is organised as follows. In Section 1.1, we describe the vision system which generated the data from observing the dice game. We also briefly describe the Progol and HR systems. In Section 2, we describe the experimental setup, including the rules of the game, and how HR and Progol were employed for this application. In Section 3, we present the results from our experiments with the vision data.

1.1 The Vision System

The vision system used in this work is composed of a video camera observing a table top where a dice game is taking place. Figure 1 shows a picture of this setup, and also depicts a schema of the system modules.

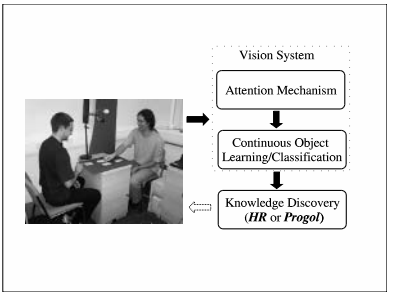


Fig. 1. A scheme of the experimental setup

Visual information obtained by the vision system is turned into a symbolic description of states of the objects observed on the table top. This is used in turn as input data for the knowledge discovery module (i.e., HR or Progol here). Outputs from the knowledge discovery module are further used to drive a virtual agent to play the observed game.

To turn video streams into symbolic information, the vision system uses a spatio-temporal attention mechanism and an object classifier. In brief, the attention mechanism uses motion as the cue to select interesting portions of space-time. Based on a generic blob tracker [7] , this mechanism works on the principle of multi-modal background modelling and foreground pixel grouping. Thus, the bounding box, centroid location and pixel segmentation are extracted from any moving object in the scene

in each frame of the video sequence. The attention mechanism identifies key frames where there is qualitatively no motion for a number of frames, which are preceded by a number of frames containing significant motion. Each object in the selected frames is classified in two steps. First, features are extracted using banks of wavelets. Second, a set of example feature vectors is partitioned into classes using a graph partitioning method. The resulting partitions are used as supervision by a conventional statistical learning algorithm that provides models. In effect, for each object identified by the attention mechanism, a symbol is associated according to the specific model in which this object is classified. For example, the statement: $state([a, b], t_{20})$ indicates that there are two distinct objects in the scene, at time t_{20}, represented by the feature classes a and b respectively. A predicate $successor/2$ representing temporal succession is also provided for each subsequent pair of time points that appear in the argument of statements $state/2$. Atoms $successor(t_i, t_j)$ state that the time point t_i is a *successor* of the time point t_j. So, the vision system may output the following symbolic representation of the scenario: {state([a], t_{307}). successor(t_{309}, t_{307}). state([a, b], t_{309}).} This indicates that the system has observed a single object of class a on the table at time t_{307}, and the next recorded event was at time t_{309}, which was when there were two objects on the table, one of class a and the other of class b respectively. The vision system is described in more details in [12].

1.2 Progol and HR

For the experiments described here, we used CProgol4.5 [11], which is an implementation of the *definite modes language* [10]. This system allows the generalisation of a set of positive examples without the need of negative examples to guide the search. This characteristic suits well our aim to explain passive visual observation, since, in this case, negative examples cannot be easily input without supervision.

In brief, Progol works as follows. For each positive example, it generates a most specific Horn clause constructed according to user defined mode declarations. Mode declarations in Progol account for restrictions in the possible form of the proposed generalisations and hence enable the imposing of language biases on the search space. The initial most specific clause is further contrasted with the remaining examples in the search for a more general formula capable of subsuming most of the dataset. The selection of this formula is made according to statistics about its coverage [11]. The selected formula is added to the Progol's clause base, while the examples that were made redundant by this formula are retracted and the inductive process is repeated for the remaining examples until the last clause is reached. Therefore, Progol outputs the generalising rules in decreasing order of statistical significance.

In contrast to Progol, the HR system performs descriptive induction to form theories about given datasets using background information which includes a set of concepts and a set of axioms which relate the concepts. It has mostly been applied to mathematical domains where it has been used to make some interesting discoveries [2].

At the heart of HR's functionality is an ability to form new concepts from old ones, with the original concepts being supplied as background knowledge. HR uses 15 production rules to perform concept formation [2], each of which is either binary – taking two old concepts as input – or unary – taking one old concept as input, and each

production rule generates a clausal, range restricted, definition of a new concept. Each production rule is parametrised so that the input set can lead to the production of multiple concepts. Five important production rules are: the *compose* production rule, that takes the definitions of two concepts and combines them by conjoining the two sets of literals in their definitions; the *match* production rule, which takes a single old definition and unifies variables to produce a new one; the *exists* production rule introduces existential quantification by changing ground values to variables in a definition; the *split* production rule, performs instantiation in definitions; and, finally, the *negate* production rule, that negates literals within definitions.

Each induced concept is assessed using a variety of measures of interestingness [3], and this drives a heuristic search through the space of definitions. In addition to generating the definitions of new concepts, HR also calculates the success sets of the definitions and uses this data to induce hypotheses about the concepts empirically. In certain domains HR employs a third party automated theorem prover (usually Otter [9]) to attempt to prove that some of the hypotheses follow from a set of axioms supplied by the user. In other domains, HR may use the theorem proving functionality to show that a hypothesis can be proved from first principles given only some simple axioms. Such hypotheses are unlikely to be interesting to the user, hence HR uses theorem proving as a filter for dull conjectures.

New functionalities, allowing it to handle noisy data have recently been implemented in HR. As these processes were essential for this application to learning from vision data we provide a brief overview as follows. Whenever HR adds a new concept to the theory, it attempts to make certain types of conjecture involving the new concept. In particular, it attempts to make implication conjectures by finding concepts where the success set is a proper subset or superset of the success set for the new concept, and formulating the hypotheses accordingly. Previously, this was restricted to cases where the subset/superset relation was *exact*. This was because, in mathematical domains, a conjecture which is 99% true empirically is normally 100% false, as a counterexample is evident in the data. However, in non-mathematical domains, such conjectures are usually interesting, and the mis-matches may be due to noise or omissions in the data, rather than indicating the falsity of the hypothesis. Hence, we have relaxed the 100% constraint, and HR is now able to make *near-implications*. That is, given a user-specified percentage P, HR will add an implication to the theory if it is supported by at least $P\%$ of the data relating to the implication.

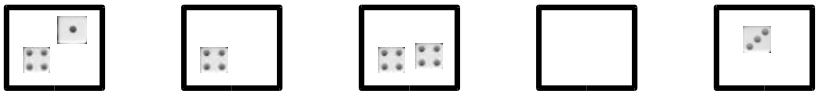
2 Experimental Setup

Eight distinct sessions where a human played the game (described in Section 2.1 below) were observed by the vision system. This produced eight separate datasets, which were used as inputs to the Progol and HR systems. Details of how Progol and HR were set up to perform the learning task are given in Sections 2.2.

2.1 Rules of the Dice Game

The dice game scenario assumed in this paper is described as follows. Two dice are thrown on an initially empty board. The game consists of keeping on the table the die

with the highest face value while the other die is replayed. Both dice are withdrawn from the table when their faces show the same figure. Some example rounds of this game are shown in Figure 2. Typical outputs of the vision system are represented in the subfigure captions, where a, b and d are constants for dice faces, [ ] represents an empty state (no dice on the table), and t_i ($i \in [1, \ldots, 5]$) are time points[1].



(a) state([a,d],t1). (b) state([d],t2). (c) state([d,d],t3). (d) state([ ],t4). (e) state([c],t5).

Fig. 2. Five states of the dice game and their typical representation output from the vision system

Given no knowledge about the numerical value of the die faces, the rules of the game can be expressed most succinctly using three general rules:

- An empty board is followed by a board with one die on it;
- A board with one die, D, on is followed by a board with D and another die on;
- A board with two equal dice faces showing is followed by an empty board;

and fifteen specific rules relating to the ordering of dice faces, as schematised by the following:

- A board with dice faces f and g showing is followed by a board with a single die with face g showing iff g $>$ f.

2.2 Setup for Progol and HR

In order to get the best performance of Progol, we used its capability to deal with lists. Therefore, subsequent states in each dataset were re-written into a predicate representing the transition between pairs of states, $trans/4$. For instance, the sequence of states represented in Figure 2 would be re-written as:

$$trans([a,d],[d],t1,t2), trans([d],[d,d],t2,t3),$$
$$trans([d,d],[\,],t3,t4), trans([\,],[c],t4,t5).$$

Given that all 18 game rules we are interested in can be expressed in terms of a transition from one board state to another, this translation enabled us to use Progol in a single run to learn a set of rules for predicting such transitions.

[1] It is worth pointing out that the assignment of dice faces to letters of the alphabet in lexicographical order, as shown in Figure 2, is for clarity purposes only; in fact, there is no preprocessing stage dedicated to this task, as the classification process assigns (consistently) any symbol to any particular scene object.

Table 1. Details of observations made by the vision system

Dataset	Number of states	Number of faces	Mis-categorisations (%)	Coverage
1	55	72	25	0.47
2	76	99	4	0.73
3	69	96	25	0.67
4	108	147	8	0.73
5	58	84	12	0.47
6	72	96	8	0.47
7	61	27	7	0.47
8	61	78	11	0.53
all	560	699	12	0.93

As with Progol, some manipulation of the symbolic information provided by the vision system was required by HR. HR cannot yet handle list representations, so the state predicates such as *state([a,c],t1)* were split into three sets of background predicates. These were: (i) *0state/1* predicates, which indicate time points when the board was empty, (ii) *1state/2* predicates, which indicate the single object on a board at a particular time, and (iii) *2state/2* predicates, which indicate the pair of objects on a board at a time point. In addition to the three predicates indicating states of boards, HR was also supplied with the *successor/2* predicate from the vision system.

We could have used the representation assumed for HR within Progol. However, in this case, Progol would not be capable of finding all the game rules in a single run, since different mode declarations would be needed. Therefore, we decided to compare the systems assuming the best domain representation respecting their potentialities and limitations.

3 Results

The observations from the vision system over eight distinct sessions are summarised in Table 1. In particular, we recorded the number of faces that were shown and the proportion of times a die face was mis-categorised by the system. We also recorded the proportion of all possible transitions from two faces to one face that occurred during each session and were correctly recorded (the column *coverage* in the table). For instance, if in a particular session, a board with dice faces a and d showing was followed by one with just die face d showing, this is counted in the *coverage* of 2-to-1 transitions observed during the session. This information is used to assess whether the sensitivity results obtained were due to the actual performance of the systems or were the effect of incomplete information in the datasets. Finally, details of a dataset with *all* the observations is given, as this was also used in learning sessions.

3.1 An Example Progol Theory

In the session described here, Progol was supplied with the vision data from dataset 2, represented as described in Section 2.2 above, and appropriate mode declarations. After

a single run, Progol found rules (1) to (19) – given below in the order produced by Progol – to solve the learning problem. Note that the symbols A and B are variables over dice faces; C and D are variables over time points, and a, b, c, d, e and f are constants encoding the emergent classes representing the dice faces.

(1) $trans([\,],[A],C,D) :- successor(D,C).$
(2) $trans([A],[A,B],C,D) :- successor(D,C).$
(3) $trans([A],[B,A],C,D) :- successor(D,C).$
(4) $trans([b,a],[b],C,D).$
(5) $trans([A,B],[\,],C,D) :- successor(D,C).$
(6) $trans([f,d],[f],C,D).$
(7) $trans([c,f],[f],C,D).$
(8) $trans([a,f],[f],C,D) :- successor(D,C).$
(9) $trans([f,c],[f],C,D) :- successor(D,C).$
(10) $trans([c,e],[e],C,D).$
(11) $trans([e,a],[e],C,D).$
(12) $trans([e,f],[f],C,D) :- successor(D,C).$
(13) $trans([d,f],[f],C,D) :- successor(D,C).$
(14) $trans([c,a],[c],C,D).$
(15) $trans([c,d],[d],C,D).$
(16) $trans([f,a],[f],C,D) :- successor(D,C).$
(17) $trans([f,b],[f],C,D).$
(18) $trans([f,e],[f],C,D) :- successor(D,C).$
(19) $trans([e,b],[e],C,D).$

Rules (1) to (19) include most of the 18 rules of the dice game we identified in Section 2.1. Rules (1), (2) and (3) correctly represent two of the three general game rules we were looking for. In particular, rule (1) represents the fact that the subsequent state of an empty state is a state with only one object (one die face). Rules (2) and (3) represent the game rule that from a state with one die on the table, a state with two dice always follows. From this dataset, Progol did not find the general rule that a board state with two equal dice faces showing is followed by an empty board. Instead, it produced rule (5), which is an over generalisation of the expected rule due to noisy data. Rules (4) and (6) to (19) represent the ordering between the faces of the dice. In total, Progol reproduced 13 out of the 18 game rules we were looking for, hence scored 13/18 = 0.72 for sensitivity.

The same method applied to dataset 3, however, only produced a sensitivity value of 0.61. Part of the problem was due to the over generalisation of rules due to noise. In fact, according to table 1, dataset 2 had 4% of miscategorisation whereas dataset 3 had 25%. Therefore, Progol shows a gentle degradation with respect to noise as it managed to produce over 60% of the rules expected even when 25% of the data was corrupt, whereas it produced 72% of expected rules for a 4% noise-corrupted dataset.

It is worth noting that Progol constructed some of the rules above using the predicate $successor/2$, whereas it ignored it in some others, where probably there were fewer examples to support them. It can be argued that the relation representing the succession of time points could be dropped in this dataset as it is implicit in the representation of transitions as $trans/4$ relations. $successor/2$ statements were kept in the representation mainly in order to differentiate rules about transitions between pairs of states from rules about the transition between longer sequences of states, which are currently under investigation.

3.2 An Example HR Theory

The theory produced on running dataset 2 by HR contained 544 implicates, of which 151 were proven to follow from the axioms supplied to Otter (as described in Section 2.2). Such proved theorems were not shown to the user, and hence only 393 were presented. The following rules in the theory were identified as relating to the 18 game rules:

```
state(A,[C]) :- successor(A,B), state(B,[nil]).
state(A,[nil]) :- successor(A,B), state(B,[C,C]).
state(A,[b]) :- successor(A,B), state(B,[b,a]).
state(A,[d]) :- successor(A,B), state(B,[d,a]).
state(A,[e]) :- successor(A,B), state(B,[e,a]).
state(A,[f]) :- successor(A,B), state(B,[f,a]).
state(A,[d]) :- successor(A,B), state(B,[b,d]).
state(A,[e]) :- successor(A,B), state(B,[b,e]).
state(A,[f]) :- successor(A,B), state(B,[b,f]).
state(A,[e]) :- successor(A,B), state(B,[e,d]).
state(A,[f]) :- successor(A,B), state(B,[f,d]).
state(A,[f]) :- successor(A,B), state(B,[e,f]).
```

As these cover 12 of the 18 game rules, HR scored 12/18 = 0.67 for sensitivity. It is interesting to note that, as in the large majority of sessions, HR found the general result that a board state with two equal faces showing is followed by an empty board state, whereas in the majority of sessions, Progol did not find this. This is probably due to the different ways Progol and HR explore the hypothesis space, while the former tries to generalise most of the data given from each unexplained example (thus becoming subject to over-generalisations), the latter constructs conjectures according to some previously defined production rules.

Conversely, while Progol often found the general rule that a board state with one face showing is followed by a board state with that face showing, plus a second one, HR never found this rule. Theoretically, HR should have found this rule, and we are currently investigating its failure.

Note also that in this session, HR did not find the rule that a board state with face b and face f showing is followed by a board state with just face f showing, whereas Progol identified this as rule, (17) above. This was due to an interesting anomaly in the data: whenever face b was showing with something else, it was always showing with face f. Thus, when HR tried to invent the concept of board states with both b and f showing, it was conjectured to be equivalent to the concept of board states with b and *anything* else showing, hence the new concept was not allowed into the theory. This had the eventual effect that the rule we expected did not appear in the theory produced.

In other sessions, where the vision mis-categorisation rate was higher, like Progol, HR also produced false results such as:

```
state(A,[a]) :- successor(A,B), state(B,[a,b])
```

These were produced because they were above the plausibility threshold of 50%. In the larger datasets, however, such incorrect rules were less common. Also like Progol, HR produced some over- generalised rules such as this:

```
state(A,[D,E]) :- successor(A,B), state(B,[C,D])
```

which is the conjecture that a 1-object board is followed by a 2-object board, but doesn't refer to the retention of the object on the board.

3.3 An Analysis of the Results

To gain an impression of the sensitivity of our approaches, we ran Progol and HR as described in Section 2.2 for each of the eight vision observation datasets, and a ninth dataset consisting of all the data taken over the eight sessions[2]. We then recorded the

[2] The percentage of mis-categorisation for this ninth session (shown in the fourth column of Table 3.1) is a weighted mean value of the mis-categorisations of its compound datasets.

proportion of the 18 rules identified in Section 2.1 which were found in the output from the systems. We then repeated the session with HR, but using a split-first search, where the split production rule is used greedily before any other. We ran a third session with HR using a compose-last search, where all production rules except compose are used greedily. In Table 2, we have recorded the sensitivity results from these sessions, along with the coverage of 2-state to 1-state board sequences as described above.

Table 2. Sensitivity analysis for HR and Progol on 9 datasets

Dataset	Coverage	Progol	HR (breadth first)	HR (split first)	HR (compose last)
1	0.47	0.44	0.44	0.44	0.44
2	0.73	0.72	0.67	0.67	0.67
3	0.67	0.61	0.61	0.56	0.61
4	0.73	0.67	0.67	0.67	0.67
5	0.47	0.5	0.33	0.44	0.33
6	0.47	0.44	0.44	0.44	0.44
7	0.47	0.44	0.39	0.44	0.39
8	0.53	0.5	0.5	0.5	0.5
all	0.93	0.83	0.83	0.83	0.83

With respect to sensitivity, Table 2 shows that HR and Progol perform as well as could be expected on the particular learning task we set, since the sensitivity obtained was always in the same range of the coverage of the dataset. Hence, as 15 of the 18 game rules were about particular sequences of boards, we could not expect Progol or HR to score much more than the coverage measure provided in Table 1 (and also reproduced in Table 2).

It is worth noting (from Table 2) that, for the task given, altering parameters for HR's search does not alter its sensitivity significantly.

We also analysed the selectivity and the efficiency of the systems. In both accounts, Progol performed better than HR. For instance, the number of rules produced by Progol for dataset 1 was 17, whereas HR produced nearly four hundred. In general the size of HR's theories were an order of magnitude larger than Progol's. Similarly, HR was more than an order of magnitude slower than Progol.

Both systems produced a number of false rules owing to noise in the data. While these do not reflect correct facts about the board game, they would not affect the way in which an agent would play the game, since the agent chooses the most statistically relevant rules first. Thus, in general, rules generated from noise are lower ranked with respect to true rules of the game (since these cover only few examples) and, thus, these do not interfere with the agent's actions.

4 Concluding Remarks

We have presented and motivated the problem of learning rules of a dice game from vision data, in the larger context of the requirement for autonomous agents to learn understandable, logical, rules about protocol behaviours. We have presented a predictive

ILP approach to this task using the Progol system and a descriptive ILP approach using the HR system, and we have compared and contrasted the systems in terms of their sensitivity when learning a particular required rule set. We conclude that, although Progol outperformed HR in most sensitivity results, the margin of difference does not allow us to affirm categorically that the former is a more suitable tool than the latter to the task given. Moreover, contrasted with the coverage of the observation, we can say that the two systems perform as well as could be expected given the data, since their selectivity was in the same range as the coverage.

Taking into account the selectivity and efficiency of the systems we can conclude that assuming a predictive approach for learning protocol behaviour from observations (following the footsteps described in [12]) is preferable in contrast to using a descriptive tool. However, the learning task proposed may have been more suitable for Progol than for HR, since the format of the rules sough were known *a priory* and every rule of the game could be found in a single run of Progol. In the general case of learning protocol behaviour, where there is no specific task to observe, the system does not have a well-defined learning task. As a consequence a predictive learning tool would be of no use, because it would not be possible to specify a single categorisation problem from which all the relevant rules of behaviour could be found. In this case, a descriptive learning approach, such as HR, would be more appropriate, since it is not constrained to solving a particular problem and hence can find rules relating to many different aspects of the background information. Also, as it does not have to worry about predictive accuracy, it will only supply faulty rules if they are sufficiently expressed in the given data. In practise, unfortunately, such an all-encompassing approach means searching a huge space of concepts and conjectures, and this may mean that rules of interest are not found efficiently enough.

References

1. S. Colton. *Automated Theory Formation in Pure Mathematics*. Springer-Verlag, 2002.
2. S. Colton and S. Muggleton. ILP for mathematical discovery. In *Proceedings of the 13th International Conference on Inductive Logic Programming*, 2003.
3. S. Colton, A. Bundy, and T. Walsh. On the notion of interestingness in automated mathematical discovery. *International Journal of Human Computer Studies*, 53(3):351–375, 2000.
4. L. de Raedt and L. Dehaspe. Clausal discovery. *Machine Learning*, 26:99–146, 1997.
5. L. Dehaspe and H. Toivonen. Discovery of frequent datalog patterns. *Data Mining and Knowledge Discovery*, 3(1):7–36, 1999.
6. R. King, K. Whelan, F. Jones, P. Reiser, C. Bryant, S. Muggleton, D. Kell, and S. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.
7. D. R. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing*, 20(8):581–594, 2004.
8. D.R. Magee, C.J. Needham, P.E. Santos and S. Rao. Inducing the focus of attention by observing patterns in space In *IJCAI Workshop on Modelling Others from Observations (MOO 2005)*, pp. 47–52, 2005.
9. W McCune. The OTTER user's guide. Technical Report ANL/90/9, Argonne National Laboratories, 1990.

10. S. Muggleton. Learning from positive data. In *International Conference on Inductive Logic Programming*, 1996.
11. S. Muggleton. CProgol4.4: a tutorial introduction. In *Relational Data Mining*, pages 160–188. Springer, 2001.
12. C. Needham, P. Santos, D. Magee, V. Devin, D. Hogg and A. Cohn. Protocols from perceptual observations *Artificial Intelligence Journal*, 167: 103–136, 2005.
13. J. Quinlan and R. Cameron-Jones. FOIL: A midterm report. In *Proceedings of the 6th European Conference on Machine Learning (LNAI 667)*. Springer, 1993.
14. A. Stehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

Mining Intonation Corpora Using Knowledge Driven Sequential Clustering*

David Escudero-Mancebo and Valentín Cardeñoso-Payo

Department of Computer Science, University of Valladolid, Valladolid 47071 Spain
descuder@infor.uva.es
http://www.infor.uva.es

Abstract. This work presents a mining methodology designed to cope with the usual data scarcity problems of intonation corpora which arises from the high variability of prosodic information. The methodology is an adaptation of a basic agglomerative clustering technique, guided by a set of domain constraints. The peculiarities of the text-to-speech intonation modelling problem are considered in order to fix the initial configuration of the cluster and the criteria to merge classes and stopping their splitting. The scarcity problem poses the need to apply a sequential selection mechanism of prosodic features, in order to obtain the initial set of classes in the cluster. A searching strategy to select the best class among a set of alternatives is proposed, which provides useful prediction models for accurate synthetic intonation. Visualization of final classes by means of a modified decision tree brings graphical cues about contrastable prosodic information of the intonation corpus.

1 Introduction

Intonation is an important attribute of the human speech which brings relevant information about many linguistic, emotional and social aspects. Despite of its importance and of the number of different approaches which can be found in the bibliography, the huge number of factors which have an effect on intonation make its modelling a very difficult challenge (see [3] for a review). The availability of recorded speech corpora opens way to data mining techniques in order to automatically extract information and generate models of intonation. Nevertheless, the nature of the problem makes it difficult to apply conventional techniques. In this work, we introduce a knowledge driven clustering technique which outcomes useful information about some intonation aspects, such as the relevance of the features and its relation with the typical patterns of intonation.

Intonation has been a matter of interest for long time in linguistics (e.g [13] for Spanish intonation). Intonation is related to the different kinds of intonation information: linguistic information (e.g. interrogative vs. declarative sentences); emotional information (e.g. the mood of the speaker) and sociolinguistic information (e.g. social and geographical origin of the speaker). In the speech technology

* This work has been partially sponsored by Spanish Government (MCYT project TIC2003-08382-C05-03) and by Consejera de Educacin (JCYL project VA053A05).

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 360–369, 2006.

domain, the primary use of intonation has been related to the improvement of naturalness of text-to-speech systems [18]. In speech recognition tasks, intonation information could provide valuable clues to find sentence boundaries or to identify the kind of sentence [16]. Speaker recognition systems have also benefited from the inclusion of intonation information [15]. Nevertheless, the huge number of variability dimensions of intonation phenomena and the difficulty to represent them adequately, justify the lack of consensus on the best way to model intonation information. The relative importance and even the right number of factors which affect intonation is still a subject of debate in the bibliography.

Time evolution of the F0 value of a speech waveform is recognized as a valuable source of information in the intonation literature. Although the first algorithm to extract F0 appeared on the sixties, it is still a subject of improvement (see [8] for a review). Moreover, there is no overall agreement on the way F0 contours should be best parameterized from the extracted F0 magnitude. The goal of Text-to-Speech (TTS) applications is to automatically obtain a mapping between a set of prosodic features affecting the shape of F0 contours and a set of parameters representing the shape of the F0 contour. This mapping could be adequately obtained using data mining techniques to intonation corpora. In the modelling stage, a mapping is inferred from the samples in the corpus. In the prediction stage, this correspondence is applied to get the synthetic contour parameters from the prosodic features derived from the labelled text. A variety of mapping techniques can be found in the bibliography, from simplified basic rule-based systems[2] to corpus based systems approaches using Neural Networks[12], Decision Trees [19], Regression Trees [1] ....

Two main limitations affect traditional learning techniques. First, they do not provide contrastable linguistic information about the intonation movements. Second, they usually lack enough robustness to cope with data scarcity problems which, as a consequence of the high number of possible combinations of potentially important prosodic features, heavily affect feature covering capabilities of the corpora (using D prosodic factors with an average number of V values each, would require unrealistic corpus sizes for typical situations in which $D > 10$ and $V > 5$, leading to more than 10^5 different units). The scarcity problem could cause unrealistic prediction of F0 contours when the input is labelled with a combination of prosodic features rarely observed or not present at all in the corpus. This can dramatically decrease the naturalness of the synthesized speech.

In this work, we will describe a knowledge driven sequential clustering which brings enough robustness to cope with data scarcity problem and provides the core component of an intonation modelling methodology which can be successfully used in TTS applications with a high degree of speech naturalness. In section 2, we formally describe the intonation modeling problem, focusing on the goals, domain constraints and limitations which inspire the decisions to be taken for the clustering procedure, which will be described in section 3. Results and conclusions are reported in sections 4 and 6.

2 Problem Statement

In corpus based intonation modelling, the corpus is considered a set of intonation units IU_i, $C = \{IU_i, \quad 1..N\}$. Every IU_i is a pair $IU_i = (IU_i.PF, IU_i.AP)$. $IU_i.AP$ is an array of numerical values which provide the acoustic parameterization of the F0 contour of IU_i. $IU_i.PF$ is a set of numbers or strings which gather the values of the set of prosodic features which label the prosodic function of IU_i. These features capture several characteristics of intonation like accent, emotion, type of sentence or grammatical structure of the sentence, among others. The prosodic features are either manually labelled in the corpus units or, in a generation stage, are derived from text using a priori linguistic knowledge.

As far as intonation is concerned, the main goal in corpus-based TTS applications is learning the correspondence between AP and PF, given a set C of labelled intonation units. Any model providing such correspondence should adequately predict $IU_i.AP'$ given $IU_i.PF'$ in the generation stage. This set $IU_i.AP'$ of acoustic parameters could then be used to generate a synthetic F0 contour close enough to intonation contours associated to the $IU_i.AP$ of the corpus which are determined to be similar to the $IU_i.AP'$.

Any procedure designed to solve the correspondence learning problem should take into account two main goals and fulfil two main constraints:

Goal 1 (G1) Prediction of synthetic F0 contours is to be as accurate as possible for Text-To-Speech applications.

Goal 2 (G2) Results of the modelling stage should provide linguistically contrastable information.

Domain Constraint 1 (C1) Two different intonation units IU_i and IU_j, are perceptually equivalent iff $IU_i.AP \sim IU_j.AP$, provided the parameterization technique has been properly selected[9].

Domain Constraint 2 (C2) The second constraint concerns to the function of intonation and it establishes that if $IU_i.PF = IU_j.PF$ then $IU_i.AP \sim IU_j.AP$, given the parameterization technique and the prosodic features have been properly selected.

Since the solution has to assume that prosodic knowledge is not complete and that it is usually impossible to gather together in a corpus a set of instances broad enough to cover all the possible prosodic configurations, two fundamental drawbacks have to be taken into account:

Drawback 1 (D1) The ideal set of PF and AP is still an open question, so that it would be desirable to obtain, as a product, information about the number of features to use, their cardinality, and their relative importance.

Drawback 2 (D2) The high number of PF involved makes data scarcity problems a fundamental difficulty to tackle with.

Together, all these goals, constraints and drawbacks drive the design of a new clustering process, adapted from basic clustering techniques [11], which will be described in the next section. As we will show, this technique will also provide visual cues which should be useful to interpret the nature of prosodic phenomena.

3 Clustering Technique

In this section, we describe our proposal for a multilevel clustering technique, driven by a forward sequential feature selection process, which correctly solves the problem described previously. The technique is inspired by classic knowledge-based agglomerative clustering techniques [11] in combination with widely accepted feature selection techniques[20].

Constraint C1 justifies the use of clustering techniques to build sets of classes of IU_i, grouped in terms of the similarity of their APs, since these sets of classes will properly represent the typical movements of F0 contours observed in the corpus. This clustering will also provide a means to find the characteristic intonation profiles, abstracting the intrinsically high variability of the prosodic events.

The process starts building an initial classification of the IU_i from a single prosodic feature $LPF_1 = \{PF^1\}$. Each class corresponds to a given value of this initial prosodic feature PF^1. An agglomerative clustering technique is iteratively applied to this cluster using maximum similarity as the merging criterion and prediction accuracy of F0 profile as the stopping condition. The prosodic feature which gives the best overall prediction accuracy of F0 profile over the cluster is selected as PF^1. An additional prosodic feature is added to LPF_1 to construct the next set of prosodic features $LPF_2 = \{PF^1, PF^2\}$ and a new cluster is build repeating the previously described process. Again, the same criterion applies for the selection of PF^2, resembling the typical forward sequential feature selection process. The clustering process stops when all the possible prosodic features have been included into $LPF_{fin} = \{PF^i | i = 1, \ldots, N_{pf}\}$ and it results into a multilevel set of clusters, each one corresponding to an increasingly more specific set of prosodic features.

Constraint C2 implies that two different intonation units sharing the same set of prosodic features are to be in the same class, since they should be similar in a way consistent with the similarity measure used to merge classes. That justifies why we choose the set of classes induced by LPF as the initial set for any cluster level.

Since the main application of this clustering will be TTS generation (see goal G1), it is clear that the stopping condition for the agglomerative process should be related to the prediction accuracy of the clusters when used to generate F0 profiles: agglomeration should stop when the prediction results using a set of clusters after merging are worse than using the present ones.

The agglomeration still provides a correspondence between APs and PFs, if we keep track of the different values of the PF associated to a class after merging. The list $PL_j = \{PF_k | k = 1, \ldots, K_{max}(j)\}$ associated to a class C_j provides an index to it which can be used in TTS to retrieve the APs which correspond to the given sequence of PF annotated in the input text. The APs retrieved sequence will be used to generate F0 contour (G1). We call *dictionary* the set of pairs $D_k = \{(PL_j, C_j), j = 1, \ldots, N(k)\}$. A dictionary is the explicit representation of the correspondence between function of intonation ($PF \in PL_j$) and its shape ($AP \in C_j$ in the class) and bring a way to fulfil goal G2.

As the number of prosodic features increases, the data scarcity problem gets worse (drawback D2). The multilevel clustering technique provides a different clustering for every $LPF_0, LPF_1, \ldots, LPF_{fin}$ and each of them has been optimally adapted to cover the IU_i set in the corpus for a given level of detail in the set of prosodic features. Since the lists LPF_j are orderly enlarged adding the next best predicting feature at each stage, we can use the corresponding ordered set of dictionaries $LD_k = \{D_j | j = 1, \ldots, k\}$ to guide a searching strategy for alternatives to unseen (or infrequent) PF_j combinations, selecting the best predicting dictionary which subsumes PF_j (refer to [4] [7] for details)

The ordered list of dictionaries provides a way to build a decision tree which gives visual information easy to contrast that schematically represents the intonation patterns found in the corpus (see section 5). The ordered set of lists $LPF_0, LPF_1, \ldots, LPF_{fin}$ provides a ranking of importance of the various prosodic features, which besides the previous visualization capabilities provides good fulfilment of goal G2 and adequately copes with drawback D1.

4 Experimental Results

For the experimental validation of the clustering technique, we have used an intonation corpus which contains more than 700 sentences (4363 intonation units) recorded by a professional actress in studio conditions[1]. High quality F0 contours were obtained using a laringograph. Sentences has been segmented and labelled following a semiautomatic process. We selected only the declarative sentences, which represent about 95% of the whole corpus. The sentences has been segmented into different types of intonation units: intonation groups (IG), stress groups (SG) and syllables (see [13] for a definition of this units). In this study the basic unit of reference has been the SG, defined as the combination of a stressed syllable of a word plus the preceding and following one. The acoustic parameters are the control points of the Bzier curves of degree 3 fitting the F0 contours in the intonation units (more details in [6]). The following prosodic features were considered: type of sentence `typeSE` (1 value), position of the tonic syllable in the first SG `posSTiniSG` (3 values) and in the last one `posSTfinSG` (3 values), number of IGs `nIGSE` (5 values), SGs `nSGSE` (6 values), syllables `nSylSE` (6 values) and phonemes `PhonSE` (6 values) in the sentence, number of stress groups `nSGIG` (6 values), syllables `nSylIG` (6 values), and phonemes `nPhonIG` (6 values) in the IG, position of the IG in the sentence `posIGSE` (7 values), position of the SG in its IG `posSGIG` (6 values), `SGBorder` indicating the configuration of the SG, number of syllables `nSylSG` (9 values) and phonemes `nPhonSG` (6 values) in the SG, position of the stressed syllable `posSTSG` (3 values). For the experiments, the corpus was split into 3 subsets: modelling, training and testing sets.

We use the centroid to represent the samples of each class in the clusters. The Euclidean distance between the respective centroids of the classes was used as the inter-class similarity metric to guide the merging process. The prediction

[1] Gently provided to us by the research group TALP of the Polythecnic University of Catalonia, Spain.

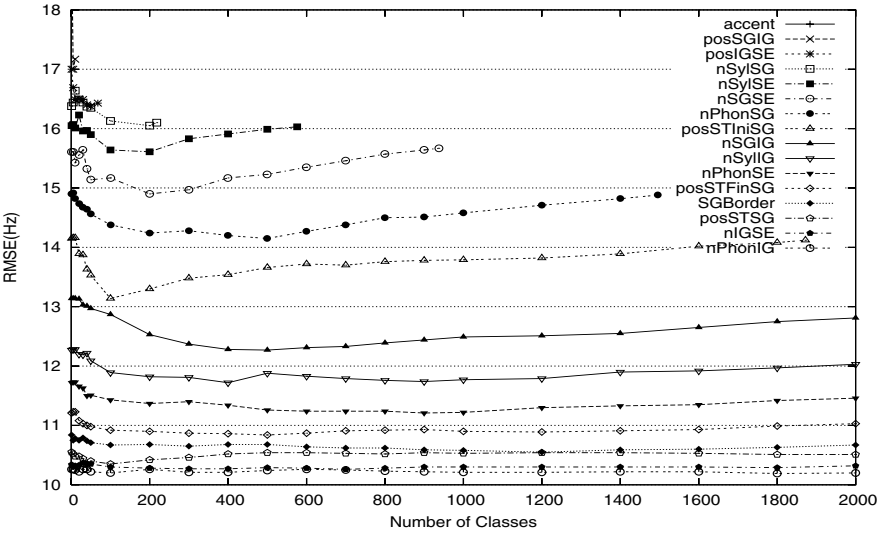


Fig. 1. Building the list of dictionaries: each curve represents the effect of adding dictionary D_i to the list of dictionaries LD_i, $(i = 1, \ldots, N_{pf})$. The name of the PF added to build D_i is the legend of the curve. Each curve represents the prediction error of the training samples as a function of the number of classes at each step of agglomeration, starting at the right end with the maximum number of classes for that set of PF. The optimal number of classes for dictionary D_i corresponds to the minimum of the associated curve.

error is computed as the distance between the points of the real F0 contour and the points of the corresponding synthetic one. This distance is measured using the recommended RMSE and Pearson Correlations [10].

Figure 1 monitors the building process of the list of dictionaries. Error values were obtained by averaging the prediction error over the set of SG in the training corpus. As the number of PF grows the impact of new PF decreases mainly due to the fact that some of the PF are redundant or they introduce few extra information (e.g. $nPhonSG$ and $nSylSG$ are correlated features: when one of them is considered, the incremental procedure rejects the other).

Prediction errors showed in table 1 indicate that the TTS results obtained using our clustering technique are comparable with other approaches found in the bibliography (see [14] for a ranking). Informal listening tests have been done to assess the goodness of the synthetic intonation. The over-training effect observed in the table could be acceptable in TTS applications where imitating the intonation patterns in the corpus does not incur any noticeable loss of naturalness.

Table 2 shows the success of using an multilevel approach. Less specific dictionaries (the ones with less number of PF) are used frequently to predict any IU, both for the testing and training corpora. The difference is more obvious for the testing set, since the number of unseen PF combinations increases for more specific dictionaries.

Table 1. Prediction Errors: RMSE y Pearson Correlation (Corr) versus the number of prosodic features in training and in testing stage. The metrics are computed using all the F0 contours of the training and testing corpus respectively.

	Train		**Test**	
List of Dict.	**RMSE(Hz)**	**Corr**	**RMSE(Hz)**	**Corr**
LD1	21.47	0.59	21.53	0.60
LD2	19.20	0.69	19.66	0.68
LD3	18.55	0.71	18.58	0.72
LD4	18.30	0.72	18.49	0.72
LD5	17.94	0.74	18.49	0.72
LD6	17.23	0.76	18.66	0.72
LD7	16.51	0.78	18.94	0.71

Table 2. Level of use of the dictionaries in a list: each cell contains the percentage of intonation units that are predicted using each of the dictionaries of the list

	Use of the Dictionary (%)						
LD7	D1	D2	D3	D4	D5	D6	D7
Train	0.0	2.8	3.4	14.8	6.1	16.1	56.8
Test	0.2	4.5	6.4	17.4	10.7	18.7	42.2

Table 3 shows the impact of the agglomerative process in the final number of representative classes: $D7$ has 1795 classes in the initial configuration and 190 at the end of the agglomeration process. Note that the initial number of classes is far away from the maximum: if the corpus would have samples to cover all the possible combinations of PF, the number of classes would be `2 (accent)` $\times$ `6 (posSGIG)` $\times$ `7 posIGSE` $\times$ `9 (nSylSG)` $\times$ `6 (nSylSE)`$\times$ `6 (nSGSE)` $\times$ `6 (nPhonSG) = 163296`. Although some of the combinations are impossible, we can easily figure out the magnitude of the corpus needed to cover them all and the importance of designing a robust strategy to cope with scarcity, as the one proposed in this work. Finally, not all the classes in the final configuration are used since this is decided through the dictionary based selection mechanism for every intonation unit: we see that only 24 out of the 190 available classes are used in $D7$. This significant reduction helps simplifying the visual representation of the clusters which will be presented in the next section.

Table 3. Description of the dictionaries in terms of the number of classes and of the number of samples per class

List of Dictionaries LD7	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
Number of eligible classes	2	4	17	30	26	21	24
Number of grouped classes	2	5	40	111	83	80	190
Initial number of nlasses	2	10	68	230	631	1068	1795
Mean number of samples per class	1235	494	113	42	35	32	16
Mean intra-class dispersion (Hz)	37	33	31	26	21	20	17

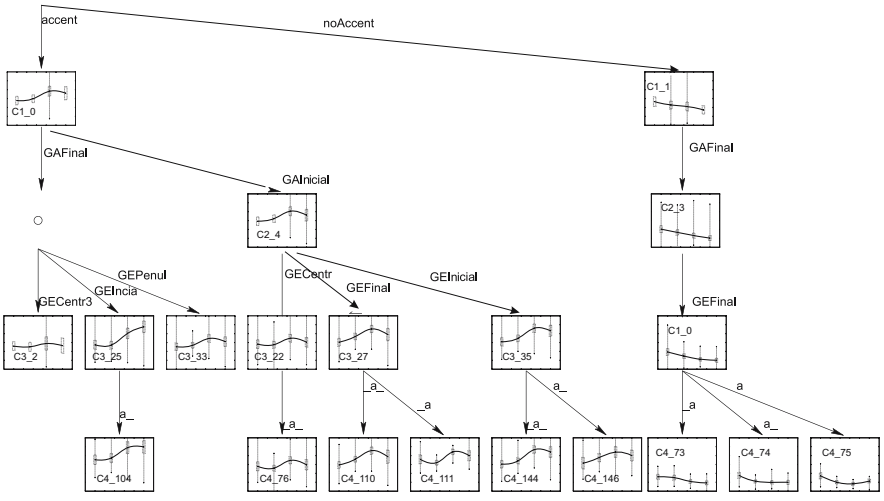


Fig. 2. Models of the dictionary represented as a decision tree. We have selected a part of the whole tree. Normalized x axis, y axis scale: 100-220Hz.

5 Visualization Cues

Figure 2 shows a tree-like graphical representation of the classes in the list of dictionaries. Each node represents a class in the clusters. For every class, we show the Bzier curve representing the F0 profile of the centroid and the standard deviation of each control point. The graph at each node provides a visual representation of the prototypical F0 patterns of the IU belonging to that class.

The classes belonging to the level i are the selected classes for dictionary D_i. Only classes which have been effectively used for prediction and contain more than 10 samples have been represented. The labels of tree branches give the values of the PF. The path going from the root to a given node provides one of the sequences of prosodic features which correspond to the node class.

This tree representation differs from a conventional regression tree in many aspects. Here the same class could appear in different nodes if more than one PF combination indexes it. Furthermore, the parent-child relationship does not imply the splitting of the samples of the parent node. Here the hierarchy is determined by the PF and the contents of the nodes by the agglomerative process. The tree is an easy to read representation of the information of the dictionaries.

The visualization of the information in the tree allows us to contrast some of the assessments found in the bibliography about Spanish Intonation. In [5], an overview of the proposals of several authors can be found. Here we review the main assessments and we contrast them with plots in figures 2.

– **The importance of the prominence.** We have labelled this function with $PF =$ `accent`. Figure 2 shows that this feature is essential: it is in the top of graph separating two sets of classes clearly different. Patterns with `accent` property, are characterized by high F0 values and by a rising pattern.

- **Position of the stressed syllable:** Prototypical patterns associated to the Spanish stress groups are $L*+H$ and the less frequent $H*$ (using TOBI notation). To analyze this fact, `nSilGA` has 4 possible values: `_a_, _a, a_, a`), where `_` means un-stressed syllable and `a` means stressed one. In this context, F0 contour evolution can be easily aligned with respect to the stressed syllable. The majority of the classes fit with the `L*+H` symbol and some of them with $H*$ according to the observations of [17].
- **Influence of the juncture:** Patterns in the *IU* boundaries have a decreasing trend (node C3_21) of the tree 2), *anticadence* (node C3_25) and *semicadence* (node C3_33) (see [13]).
- **Type of sentence:** affecting mainly the last part of the F0 contour. Typical final juncture of declarative sentences $L*+L\%$ is clearly seen in figure 2.

Finally, we remark that the visualization of figure 2 will probably let the experts to conclude about the intonation phenomena, although a thorough discussion of this is out of the scope of the present paper.

6 Conclusions and Future Work

The peculiarities of the intonation modelling problem have inspired the definition of an ad-hoc clustering methodology. The methodology provides synthetic F0 contours of a comparable quality to the ones found in the state of the art.

The methodology does not depend on the selection of the prosodic features, acoustic parameters, and type of intonation unit. This could be exploited to experiment the effect of those prosodic factors on quality intonation modelling.

Extracting contrastable information from the corpus of study was also a goal, in a field where many conceptual questions are still open. The proposed clustering procedure provides a ranking of importance of the prosodic features typically used to classify intonation patterns.

Furthermore, the tree-like representation of the result classes provides visual cues which aid contrast relationships between prosodic features and typical F0 contours patterns, as found in the working corpus. This information could be most valuable as an objective means to test the intonation of a given corpus against others or to validate the linguistically correct intonation of a given corpus, with respect to a set of recognizable theoretical prosodic assessments. The results presented for Spanish show good agreement with accepted prosodic knowledge for this lenguage.

References

1. P.D. Aguado, K. Wimmer, and A. Bonafonte. Joint extraction and prediction of fujisaki's intonation model parameters. In *Proceedings of Eurospeech 2005*, 2005.
2. J. Allen, M. S. Hunnicutt, and D. Klatt. *From Text to Speech: The MITalk System.* Cambridge University Press, 1987.
3. A. Botinis, B. Granstrom, and B. Moebius. Developments and Paradigms in Intonation Research. *Speech Communications*, 33:263–296, July 2001.

4. V. Cardeoso and D. Escudero. A strategy to solve data scarcity problems in corpus based intonation modelling. In *Proceedings of ICASSP 2004*, 2004.
5. D. Escudero. *Modelado Estadstico de Entonacin con Funciones de Bzier: Aplicaciones a la Conversin Texto Voz.* PhD thesis, Dpto. de Informtica, Universidad de Valladolid, Espaa, 2002.
6. D. Escudero and V. Cardeoso A. Bonafonte. Corpus based extraction of quantitative prosodic parameters of stress groups in spanish. In *Proceedings of ICASSP 2002*, Mayo 2002.
7. D. Escudero and V. Cardeoso. Optimized selection of intonation dictionaries in corpus based intonation modelling. In *Proceedings of Eurospeech*, September 2005.
8. D. Gerhard. Pitch extraction and fundamental frequency: History and current techniques. Technical Report TR-CS 2003-06, Department of Computer Science, University of Regina, Regina, Saskatchewan, CANADA, November 2003.
9. J. Hart, R. Collier, and A. Cohen. *A perceptual study of intonation. An experimental approach to speech melody.* Cambrige University Press, 1990.
10. D. J. Hermes. Measuring the perceptual similarity of pitch contours. *Journal of Speech, Language, and Hearing Research*, 41:73–82, February 1994.
11. A.K. Jain, M.N. Murty, and P.J.Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, September 1999.
12. O. Joskisch, H. Mixdorff, H. Kruschke, and U. Kordon. Learning the parameters of quantitative prosody models. In *Proceedings of ICSLP 2000*, 2000.
13. T. Navarro-Toms. *Manual de Entonacin Espaola.* Madrid, Guadarrama, 1944.
14. S. Sakai. Additive modeling of english f0 contours for speech synthesis. In *Proceedings of ICASSP 2005*, 2005.
15. E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke. Modeling Prosodic Feature Sequences for Speaker Recognition. *Speech Communication*, 46 (3-4):455–472, 2005.
16. E. Shriberg, A. Stolcke, D. Hakkani, and G. Tur. Prosody-Based Automatic Segmentation into Sentences and Topics. *Speech Communication*, 32(1-2):127–154, 2000.
17. J. M. Sosa. *La Entonacin del Espaol.* Ctedra, 1999.
18. R. Sproat. *Multilingual Text-to-Speech Synthesis.* Kluwert, 1998.
19. P. Taylor. Analysis and Synthesis of Intonation using the Tilt Model. *Journal of Acoustical Society of America*, 107(3):1697–1714, 2000.
20. A. Webb. *Statistical Pattern Recognition.* Wiley, 2nd edition, 2002.

Using Common Sense to Recognize Cultural Differences

Junia Anacleto[1], Henry Lieberman[2], Aparecido de Carvalho[1], and Vânia Néris[1], Muriel Godoi[1], Marie Tsutsumi[1], Jose Espinosa[2], Américo Talarico[1], and Silvia Zem-Mascarenhas[1]

[1] Advanced Interaction Laboratory - LIA
UFSCar - Rod. Washigton Luis KM 235 - São Carlos - SP - Brazil
{junia, fabiano, vania, muriel_godoi, marie_tsutsumi, americo}@dc.ufscar.br; silvia-zem@power.ufscar.br
http://lia.dc.ufscar.br
[2] MIT Media Laboratory
20 Ames St., 384A - Cambridge - MA - 02139
{lieber, jhe}@media.mit.edu
http://www.media.mit.edu

Abstract. This work focuses on evaluating whether cultural differences can be recognized in knowledge bases that store common sense. We are studying this issue using knowledge bases in different languages that contain thousands of sentences describing people and everyday activities, collected from volunteer Web contributors, in three different cultures: Brazil, Mexico and the USA. We describe our experiences with these knowledge bases, and software which automatically searches for cultural differences amongst the three cultures taking into account the eating habits of those cultures, alerting the user to potential differences. Though preliminary, we hope that our work will contribute to software that takes better account of such differences, and fosters inter-cultural collaboration.

Keywords: common sense, cultural differences, knowledge bases.

1 Cultural Knowledge in User Interfaces

Many researchers have pointed out that cultural differences must be considered in the design of interactive systems [10,4]. User-interface developers may need to go to "culture class" in order to minimize culture clashes in their designs [10]. Individuals communicate with each other by assigning meaning to messages based on their prior beliefs, attitudes, and values [4]. Culture clashes in interfaces occur when a user cannot interpret an interface element because the user does not share the underlying assumptions used by the designer to convey a message. For example, the "desktop" metaphor of files and wastebaskets may not make sense in a culture where work does not routinely occur at desks.

Research in culturally-sensitive design has largely taken the form of admonitions to designers to be aware of cultural characteristics and consider them in designs, and to test designs with users of the intended target culture. However, there are

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 370 – 379, 2006.

limitations to this approach. Since cultural knowledge is often implicit, designers often have trouble even realizing that their designs carry implicit cultural dependencies. And we can't design by hand for every combination of possible cultures, nor is it always practical to exhaustively test for every possible user culture. An alternative, ambitious as it might seem, is to try to actually represent cultural knowledge in the machine, and have interfaces that automatically and dynamically adapt to different cultures. While fully implementing this goal is still out of reach, this paper takes some first steps.

We have collected large knowledge bases representing common sense knowledge in three cultures: Brazil, Mexico and the USA. Comparison between these knowledge bases gives us a basis for automatically discovering differences between cultures, and finding analogies from one culture to another. Software for cultural comparison is useful in many contexts. For example, by those who want to develop systems focusing on a specific user group (e.g. a teacher who consults the common sense database to prepare a specific instructional content); by those who want to develop systems which use the cultural knowledge stored in the knowledge bases (e.g search engines that consider the context); and by those who want to facilitate communication between people, providing mutual knowledge about their cultures.

In this paper we share our experience with what we have found in cultural comparison, focusing on the domain of food, cooking and eating habits. We have also implemented a prototype software agent that watches user input and alerts the user to possibly relevant cultural differences and analogies.

It is important that the reader understands that we make no claim to have fully described a national culture, or to have fully captured cultural differences. Instead, the goal is simply to get some useful representation of culture and common knowledge, where otherwise the computer would have none. Of course, there will always be many subtle issues between cultures that will be missed or incorrectly identified by software such as ours, once we are adopting a fail-soft approach.

We believe that our contribution is the first attempt to systematically study the extent and nature of cultural differences; to represent cultural differences in machine-readable form; and to present an example of software that searches for such differences and provides inter-cultural translations automatically. In any event, the attempt to do so should shed light on the problem and increase inter-cultural awareness.

The next section presents how data are collected in the Open Mind Common Sense (OMCS) knowledge bases. To give the reader some idea of how knowledge in the various databases compares, we present some example comparisons for the food domain. We then discuss the prototype agent for finding cultural differences, followed by conclusion and future work.

2 The Open Mind Common Sense Approach for Gathering and Using Common Sense Facts

One widely accepted definition to common sense is the knowledge that people typically assume is possessed by ordinary people in a given culture. One way that AI has used to represent this knowledge is by simple sentences asserting such facts (pioneered

by Douglas Lenat's CYC project). “Ice is cold” and “If you forget someone’s birthday, they may be unhappy with you” are samples of those sentences. Common sense knowledge, thus defined, spans a huge portion of human experience, encompassing knowledge about the spatial, physical, social, temporal and psychological aspects of typical everyday life.

The idea behind Push Singh and colleagues' Open Mind Common Sense site (http://opemind.media.mit.edu), which asks the general public to input sentences expressing common sense knowledge in natural language [11,6], is that every ordinary person has the common sense that computers lack, so they can help to build the knowledge base that is necessary to give computers what they need to be able of common sense reasoning.

In this manner, after registering and logging onto the system, users are put in contact to several activities that are proposed to gather different kind of common sense knowledge. Some of those activities are template-based and others allow the user to provide entries in free form language, what can be observed in Figure 1. Filling out the activities results in growing the common sense facts database. Since 2001, approximately 800,000 sentences have been collected in the English-language version from 20,000 users.

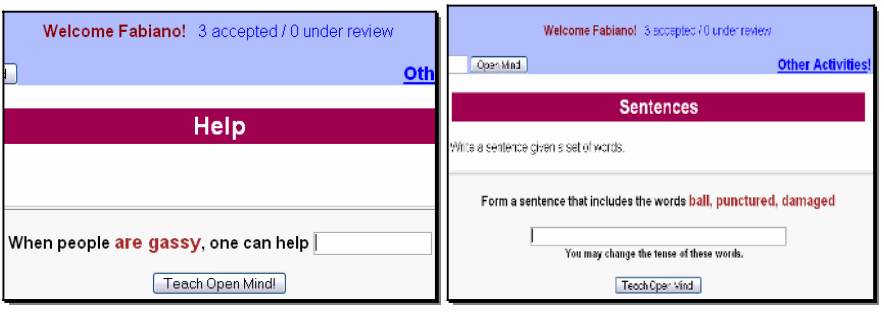


Fig. 1. Template-based and free form activity example

The data are stored in the Open Mind Common Sense database as simple statements in natural language. However, for machine use, it is necessary to put them in a representation that allows machines to make practical inferences and analogies. For that, the data are submitted to a natural-language parser that generates a set of normalized nodes that are semantically related, composing a semantic network, which is called ConceptNet. A better understanding about how this semantic network is generated is presented by Liu [7].

Common sense is acquired from the interaction with the environment. Changing the environment changes the perception of common sense and is one of the reasons why different and diverse cultures exist [7]. We set out to discover how common sense knowledge is different in different cultures. So, the group in Brazil, translated the OMCS site itself (not the existing English knowledge base) into Portuguese, and

launched the first non-English version, (http://sensocomum.ufscar.br), to tremendous response. In the span of a few months, it has acquired about 102,000 sentences.

A Spanish site was recently launched, (http://openmind.fi-p.unam.mx), but for this project, a small subset of knowledge for Mexico was created by a native Mexican. A Japanese site will be launched by the time of the conference, and sites are under development in Korean, Chinese, and other languages. With common sense knowledge bases in different languages we are now in a position to ask questions as, "How much difference is there between different cultures?", "Where do cultures agree and disagree?" and "What topics are different cultures interested in?".

For the comparison experiments, we focused on a single, though broad, subject matter domain – food. Food is a good topic because it is known to be culturally dependent, and is often the subject of social interaction. We found out that roughly 20% of the content of the Brazilian site concerned food and associated topics – it is obviously a topic that is important in people's minds! Again, though, our focus on food is not to preclude other topics – we expect the methodology to be valid for the full range of common sense.

3 Recognizing Cultural Differences

As it was mentioned before, the main purpose of this work is to evaluate whether cultural differences can be recognized in knowledge bases that store common sense. For that, we selected a theme which frequently appears in the Brazilian OMCS knowledge base, food, and used the following syllogism: taking into account that *(i)* eating habits express culture and that *(ii)* common sense involves eating habits, therefore *(iii)* common sense expresses culture. In this work, *i* is considered true and, since *ii* is verified by the analysis of the content, related to food, stored in the OMCS knowledge bases, *iii* is proved.

Here we present some eating habits aspects that we believe to be associated to cultural values. We collected and compared them from the Brazilian, Mexican and American OMCS knowledge bases.

The first step of the analysis, which led us to those aspects, was the selection of facts related to food, for example: "People eat salami when they drink beer" or "You generally want a hot dog for lunch". Considering the redundancies in our data, we selected three categories that appeared in higher frequency in each base. The categories are: (1) at what time people have their meals, (2) what people eat for each meal and (3) what kind of food people have for special occasions as parties and Christmas.

Again, the idea is not to present a definitive scientific survey about such issues as what people actually eat for breakfast in Brazil versus the USA. The knowledge, after all, comes from members of the general public who may have legitimate disagreements or differing experiences about these issues. The idea is that common sense collects plausible (rather than completely accurate) answers about these questions, which can lead to plausible assumptions about cultural differences.

Time for Meals

One of the themes that commonly appear in the knowledge bases is the time for meals. Table 1 shows what is considered common sense for most of the collaborators.

Table 1. Time for meals

	Brazil	Mexico	USA
Lunch	11:30 to 13:00	14:00 to 16:00	12:00 to 14:00
Dinner	18:30 to 20:00	20:00 to 21:00	18:00 to 19:00

Here it can be observed that meals in Mexico are the latest one. Although in Brazil and USA meals happen in similar hour in Mexico it seems to be common to have lunch or dinner about two hours later.

What Do People Eat in Each Meal?

Differences between what is eaten in each meal also can be noticed. Table 2 shows what seems to be considered common sense about what to eat.

Table 2. What do they eat in each meal?

	Brazil	Mexico	USA
Breakfast	bread	tamales, eggs with hot sauce	pancakes, bagels
Lunch	rice, beans, meat, salad, egg	chicken with mole, roast meat, pastries, chilaquiles, barbacoa, tacos	hamburger, hot dog, pizza, sandwiches
Dinner	rice and beans, soup, salad, sandwich	tamales and atole, quesadillas, coffee and cookies, bread with bean	steak and eggs, baked chicken, clam chowder, mashed potatoes
Dessert	ice cream, fruit, candy	rice with milk, churros with chocolate, nuts with honey (crowbar), sweet coconut	pumpkin pie, apple pie, ice cream, cheese cake

It is possible to notice that Brazilian people prepare lighter food at breakfast. Also Mexican people seem to like food made with flour. Concerning desserts, Brazilian people associate ice cream as something cooling, and are reluctant to eat it in Brazil's (relatively mild) winter. In the other hand, American people seem to prefer pies and other baked desserts. Chocolate for dessert is remembered in all the three cultures.

Food for Special Occasions
Christmas, and parties were topics that contributors remembered too. Table 3 shows the main types of food cited for this occasions.

Table 3. Food for special occasions

	Party	Christmas	Birthday
Brazil	snack, candy, cake, meat, beer	turkey, pork, lamb	cake
Mexico	beer, tequila	romeritos, codfish, spaghetti	cake
USA	beer, vodka	cranberry sauce, pine-apple salad, frozen Christmas pudding	cake

It is interesting to notice that in Brazil and México it seems to be common have salty food for Christmas while in USA sweet dishes seem to be more appreciated. In the other hand, beer seems to be appreciated in parties in the three countries. And also, everybody seems to like to celebrate birthday eating a birthday cake.

4 Facilitating Communication Between People by Showing Cultural Differences

In a global setting, people will have diverse cultural backgrounds provoking all benefits and challenges that intercultural communication can offer. When members can be expected to come from diverse cultures, diversity can be expected in terms of cultural styles, communication procedures, interaction, values, etc. Hence it is necessary to know what is specific about intercultural communication because different cultures have different notions of how to begin and develop conversations [13,3].

Due to the difficulties in grounding and differences in cultural norms, participants may experience less satisfaction with cross-cultural interactions than with same-culture interactions. [12]

In this work we believe that giving to computer some common sense can help to minimize such problems by showing the cultural differences during a computer-based social interaction among people.

Therefore, we present our first prototype application, “What Is He Thinking?” (WIHT), which implements an intelligent agent that provides safe-topic suggestions. WHIT’s agent continuously watches what the user is typing, while it makes commentaries on the differences in the grounding that can lead to possible misunderstandings. The system also uses these differences to calculate analogies for concepts that evoke a similar social meaning in those cultures. We focus this prototype on the social interaction among people in the context of eating habits. This domain about food was selected to exemplify the potential of the application, but it could scale to other domains.

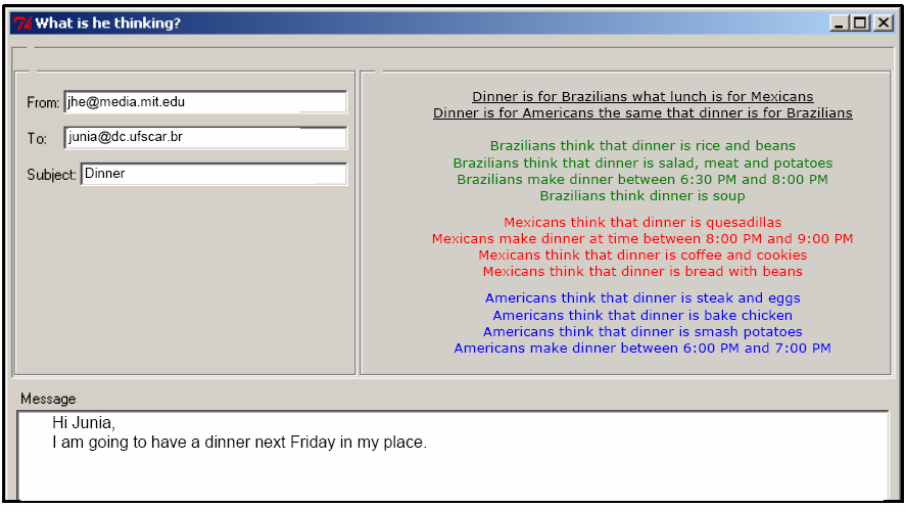


Fig. 2. A screen shot of the system

The system's interface has three sections, as can be seen in Figure 2. The first one – at the upper left – is the information for the email addresses and the subject, the second one – at the upper right – is where the agent posts its commentaries about the cultural differences and the third part – the lower part – is the body of the message.

The second section has four subsections: the upper one shows the analogies that the agent found and the other three show the data that are not suitable for analogy. For example, in our screen shot, the third label for the Mexican culture – Mexicans thinks that dinner is coffee and cookies – and the second for American culture – Americans think that dinner is baked chicken – cannot make a meaningful analogy even if they differ only in one term.

The user can click on the analogy and the system shows an explanation of the process used to construct the analogy as it is shown in Figure 3. It unfolds the steps of the inference used to generate the analogy.

Fig. 3. Explanation of the analogy "Dinner is for Brazilians what lunch is for Mexicans"

In order to make the cultural analogies, the system uses four semantic networks. The OMCSNet [9] semantic network (OMCSNet.OM), which was mined from the Open Mind corpus, is used as the core engine because it provides tools for context expansion and is especially designed for working with Open Mind Common Sense

databases. The other three databases are culturally specific; they have knowledge about the Brazilian, Mexican and North-American culture – these semantic networks are called OMCSNet.BR, OMCSNet.MX and OMCSNet.US respectively. The OMCSNet.BR was built from data mined from the Brazilian Open Mind Common Sense database. In the American and Mexican cases, the statements were already in English language. In the Brazilian case, the statements were originally in Portuguese. For this project, a small group of statements related to eating habits were selected and freely translated to English to be parsed.

The calculation of cultural analogies is divided into eight steps:

1. Data retrieval: The first thing that the system does is to use the NLP package MontyLingua [8] to get the relevant concepts of the mail. This information is presented as tuples of (verb subject direct_object indirect_object). In the example above, the NLP tool produces the output ("have" "I" "dinner" "my place").

2. Context retrieval: Then, we use each direct and indirect object of the tuples from the previous step to query the OMCSNet.OM for the relevant concepts. Querying this network first gives us some query expansion of "culturally independent" relations. That is, the base OMCSNet.OM, as our largest and most diverse collection, is taken as the "standard" to which the other databases are compared. Since OMCSNet.OM itself has some cultural bias, it would be better, once we have collected enough knowledge bases in other languages, to create a "worldwide" knowledge base by removing all culturally-specific statements, as a basis for comparison. The result of this query is used to get the context in the culturally specific nets. At the end of this stage the output was ranked using two criteria: the first one prefers the concepts resulting for the cultural databases, and the second is to rank first the concepts that come from the part of the email that the user has just written. This helps to address the relevance effectively by giving preference to the important topics of each culture, and in the recent topics of the mail. This process brings concepts as lunch, food, meal, and salad, which are in the semantic neighborhood of dinner.

3. Node retrieval: In this step, we get the tuples whose nodes are in the context of the mail from OMCSNet.BR, OMCSNet.MX and OMCSNet.US. The output of this step is the pieces of common sense knowledge for the Brazilian, Mexican and American culture (see Figure 4). At this point we have everything the databases have about the eating habits in both cultures. For example: the output from OMCSNet.MX has the following, among others: ['TakeTime', 'dinner', 'between 8:00 PM and 9:00 PM'], ['KindOf', 'dinner', 'light meal'], ['IsA', 'dessert', 'rice with milk'], ['IsA', 'dinner', 'coffee and cookies'], ['IsA' 'food' 'chocolate']; and from OMCSNet.US has: ['TakeTime', 'dinner', 'between 6:00 PM and 7:00 PM'], ['KindOf', 'dinner', 'heavy meal'], ['IsA', 'dessert', 'pumpkin pie'], ['IsA' 'food' 'chocolate'].

4. Relevance of the nodes: By comparing each node with the cultural sets of knowledge in the previous step we can get its relevance. For this operation, the SIM operation from Cohen's WHIRL [2] is used. The interesting part about WHIRL is that it is a system that effectively interleaves inference with retrieval. This operation allows getting the similarity for each node in one set with all the elements of the other set. The operation always maps to a number between zero and one.

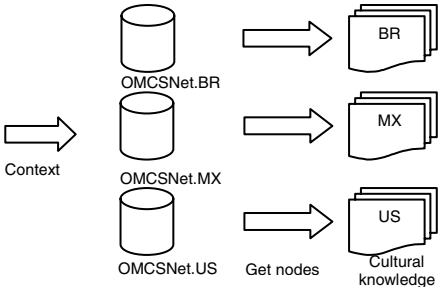


Fig. 4. The node retrieval operation

5. Calculation of analogies: If the value of one node and the semantic relations in the tuples of one set are equal to the tuples of the other cultural set, then the unmatching concept is an analogy between the two cultures that are being considered. In addition, the semantic relation is analyzed in order to avoid irrelevant analogies. These analogies are ranked using the similarity between the two nodes. If two different pair of tuples allow the same analogy, the values of the relevance scores are added. In our example set, the only nodes that are suitable for analogy are the nodes that talk about the 'KindOf' meal the dinner is in both cultures. This process is similar to Dedre Gentner's classic Structure Mapping analogy method.

6. Calculate the nodes to display: The nodes are sorted using the relevance of their context and their similarities produced by the SIM operator.

7. Chose the information to display: The nodes ranked higher are chosen to be displayed. First, we choose the nodes to calculate analogies and then the rest of the nodes, the former nodes give information about things that are different, but do not have a counterpart in the other culture, or are grounding information that makes no sense for an analogy [0]. See next step for more details.

8. Map the concept to English: For each semantic relation in the net, a custom template that maps its information to English sentences was created and applied before displaying the information. For the analogies, an additional template is used to explain why the system made this analogy.

5 Conclusion and Future Works

This work, partially supported by the TIDIA-Ae FAPESP project, proc. no. 03/08276-3, and by CAPES, has presented a preliminary analysis that points to the existence of cultural differences between the data stored in the OMCS knowledge bases. We model cultural differences by comparing commonsense statements collected from volunteer Web contributors from various cultures. These differences should be considered by those who want to facilitate communication between people, providing mutual knowledge about their cultures [12,13].

As future work we are going to perform systematic user testing. While we have not yet conducted formal user tests with the e-mail application, informal feedback from users shows that, while the absolute accuracy of suggestions is not high, users do

appreciate the occasional useful suggestion and it is not excessively distracting even when suggestions are not relevant. This is consistent with our experience in more formal user testing of our other common sense applications such as ARIA [5], Globuddy [5], and others described in the references. We hope developers of interactive systems use the knowledge about culture stored in OMCS databases in order to facilitate the interaction between humans and computers with some intelligence.

Further works will also include the investigation of cultural expressions in OMCS knowledge bases considering a larger number of facts. Other domains will also be studied in order to verify the cultural differences besides eating habits domain. Thus, more details about cultural factors and how they would affect intercultural conversations will be inquired.

References

1. Brown, P., & Levinson, S. Universals in Language Usage: Politeness phenomena. In E. Goody (Ed.), Questions and politeness (1987), p. 56-289.
2. Cohen, William W. WHIRL: A word-based information representation language. *Artificial Intelligence* (2000), p. 163-196.
3. Isbister, K - Designing an Assistant for Human-Human Interaction in a Virtual Meeting Space. Proc. CHI (2000)
4. Khaslavsky, J. Integrating Culture into Interface Design. *Proc CHI (1998)*, p 365-366.
5. Lieberman, L., Liu, H., Singh, P., Barry, B. Beating Common Sense into Interactive Applications. *AAAI* (2005), p 63-76
6. Liu, H.; Singh, P. Commonsense Reasoning in and over Natural Language. *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems. KES,* (2004). Last visited in Sept, 2005. <http://web.media.mit.edu/~push/CommonsenseInOverNL.pdf>.
7. Liu, H; Singh P. ConceptNet: A Practical Commonsense Reasoning Toolkit. *BT Technology Journal*, v. 22, n. 4, p. 211-226, (2004). Last visited in Sept, 2005. http://web.media.mit.edu/~push/ConceptNet-BTTJ.pdf
8. Liu, Hugo. MontyLingua: An End-to-End Natural Language Processor for English. (2003)
9. Liu, Hugo and Singh, Push. OMCSNet: A Commonsense Inference Toolkit. MIT Media Lab Society Of Mind Group Technical Report SOM02-01. (2002)
10. Marcus, A. Culture Class vs. Culture Clash. *Proc. Interactions* (2002), 25 p.
11. Singh, P. The OpenMind Commonsense Project. *KurzweilAI.net,* (2002). Available in: :< http://web.media.mit.edu/~push/OMCSProject.pdf>. Last accessed in September, 2005.
12. Setlock, L - Collaborating within and across Cultures in Face-to-Face Settings and via Instant Messaging. Proc CSCW (2004)
13. Zorn, I. Do culture and technology interact. Overcoming technological barriers to intercultural communication in virtual communities. SIGGROUP Bulletin, Volume 25, Issue 2, (2005).

Detection of Repetitive Patterns in Action Sequences with Noise in Programming by Demonstration

Raqueline R. de M. Penteado[1], Sérgio R. P. Da Silva[1], Alexandre H. Furuta[1], and Muriel de S. Godoi[2]

1 Departamento de Informática
Universidade Estadual de Maringá
Av. Colombo, 5790, zona 07 - Maringá – PR, Brazil - 87020-900
{srsilva, raque, furuta}@din.uem.br
[2] Departamento de Computação
Universidade Federal de São Carlos
Rod. Washington Luís, Km 235, São carlos, SP, Brazil - 13565-905
murielgodoi@gmail.com

Abstract. Software applications that exploit implicit programming by demonstration should be able to detect repetitive patterns in user's actions in an autonomous and efficient way. We present a software agent for the detection of repetitive action patterns that makes use of domain knowledge in this process. We explain its design rationale and discuss some of its advantages, by comparing it with the classic algorithm KRM, which does not make use of domain knowledge. We demonstrate that our agent might have a more efficient detection process for repetitive tasks since it activates the search algorithm fewer times. Moreover, we show that it can detect repetitive tasks even in the presence of noise in the action sequence.

1 Introduction

Users of the current interactive applications would be much more productive if the repetitive actions they execute could be automated. To deal with this problem, current commercial applications have made available mechanisms for users to create some extensions to their functionalities, like the macro recorder and script languages [1]. However, both mechanisms have the same common drawback of interrupting the users' normal flow of work. On the other hand, programming by demonstration (PbyD) brings the possibility of combining the simple interaction of a macro recorder with the expressivity power of a script language [1]. Like the macro recorder, PbyD applications allow users to demonstrate the sequence of actions they want to automate using the interface language. From this demonstrated sequence, they should generate a program that executes the user's task automatically. This way, the learned programs could have complex control structures like: iteration, conditional and abstraction [1], showing an expressive power like the ones made available in script languages.

PbyD mechanisms could be classified as explicit or implicit [12]. In both cases, an application that uses it should carry out the following steps: 1) capture the sequence

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 380 – 389, 2006.

of actions the users wants to automate by demonstration in the user interface; 2) detect the existing repetitive patterns in this sequence; 3) infer the possible generalization on top of the detected patterns; and 4) show this generalizations to the users for their approval and use in a non intrusive form [10]. In explicit PbyD, there still happens an interruption in the user's normal flow of work since they need to show, explicitly, the start and end of the action sequence they want to automate, as it happens in SmatEdit [8]. On the other hand, in implicit PbyD the sequence capture will happen in background, by an agent that monitor the user's action identifying the start and end of the sequence, without interrupting the user, as it happens in Eager [2] and APE [12].

In this paper, despite the essentiality of steps 3 and 4 in the development of PbyD applications, we will concentrate our discussion in the problems related to the steps 1 and 2, that is to say: the capture of the action sequence, the identification of the start and end of a action pattern for the work domain and the treatment of noise in the action sequence. Additional detail about our approach to the other steps could be found in [11] and [4].

Trying to minimize the problems found in steps 1 and 2, we propose the use of domain knowledge in the detection process of the repetitive patterns present in the users' actions. Our approach explores the fact that application interaction sections, from the linguistic point of view, could be seen as a *text*. More specifically, we see it as a set of paragraphs. Also, to maintain the textual cohesion, a paragraph should always refer to a unique object in the text — the *focus of discourse*. This fact greatly facilitates the precise identification of the start and end of repetitive patterns that happens in the action sequences of interaction sections, because the sequence will refer to a unique object. Furthermore, this change of perspective, from individual actions to a sequence of actions with focus on the manipulated object, allows the use of domain knowledge, in the form of a state machine that models the valid object action sequences, making it possible to treat the noise present in users' action sequences. At the same time, it will be possible to minimize the number of times the detector agent will be activated, when compared to current implicit PbyD applications

In the next section, we will describe the repetitive pattern detection process and the knowledge structure that it uses. Following, we show an experimental analysis done with our pattern detection algorithm and the KRM [6] pattern detection algorithm, used in the APE [12] application. To conclude, we discuss the results obtained and propose some future works.

2 Using Domain Knowledge in the Detection of Tasks

An implicit PbyD application should automatically capture the sequence of user actions done at the interface and detect the repetitive patterns present in it to generate the extensions that automate them. In the detection process, current PbyD application, generally, take into account each action from the sequence individually, assuming that there is no relation between them. This assumption, which consider the user interaction as a sequence of commands exchanged between s/he and the application is derived from the systemic view [5] applied to the human-computer interaction and degrades the performance of the agent that detects repetitive patterns. This happens because the agent has to verify the possibility to form meaningful action patterns with

all the existing actions of the sequence or at least with a limited set of past actions, which form the context for the patterns to occur. It is interesting to notice that although the current implementations can detect patterns in consecutive and non consecutive action sequences [11], they need the sequences to be identical, i.e., they should have the same actions and this actions should be executed in the same order. As a result, they could not tolerate the presence of noise in the sequence.

We adopt the Semiotic Engineering approach [3] as our theoretical basis. This approach considers the user interaction as a *communication* between the user and the application, using the media perspective [5] to human-computer interaction. This way, we can see a user interaction section in the application as *text*. More specifically, in this paper, we consider a user section as a set of paragraphs. Hence, using linguistics concepts related to pragmatics we verify that, to keep the textual cohesion [9], a paragraph always refer to a unique object in the text — the *focus of discourse*. As a result, generally, any action that happens in a paragraph will have as a subject this object. From these observations, we can find that the belief that the actions done by users in the interface are independent is not correct, since they are connected by the focus object on which they act.

The above point of view suggests us that we should take into account the structure of the tasks in the repetition detection process. As a result, we decide to use a state machine in conjunction with each class of a domain application ontology (an ontology that models the users' view of the application elements) to help accelerate the detection process. Thus we define a *task* as a sequence of dependent valid action on the same object, and take as *repetitive tasks* the tasks that have the same actions executed consecutively or not, even if these actions happen on different objects. For example, the sequence of actions "select and alter the font size" of a 'table' and "select and alter the font size" of a 'paragraph' that could happen in a text editor will form a repetitive task, since they have the same actions ("select and alter the font size").

From the above definitions, we could identify some important elements for modeling the state machine of a domain object. When modeling an object state machine the software designers should consider all the actions that could be executed on this object and all the valid action sequences on it. Nevertheless, they could not insert loops in the machine, once the detector agent will consider them, automatically, when it identifies repetitive tasks done in the application's interface. We still recommend that the designer consider as dependent actions only those actions where there is an unconditional dependence between them. For example, for a table to be "pasted" it necessarily should be "copied" or "cut" first.

Fig. 1 shows the state machine for the object 'word' of a simple adaptive text editor, which will be used throughout the text as a case study. It is interesting to observe that a task could be a set of dependent actions ("select, copy, and paste") as well as a unique action ("delete"). This fact is very relevant because it affects sensibly the detection process. The non identification of individual actions as valid tasks could result in the non detection of meaningful patterns. It also could make the detection process irregular and, consequently, weak, given that the necessary automation will not occur in the user's work. A typical example of this problem occurs with the action "delete" in the action sequence ("select, copy, paste, and delete") applied on a 'paragraph'. In this case, if we do not consider the action "delete" as a valid task, the detection process will be compromised, since the detector agent will consider as similar tasks

only the ones that have the whole action sequence identical, i.e., a sequence of four actions. Thus, if, after this sequence being done, a user "select", "copy" and "paste' a 'table', a repetitive tasks will not be detected immediately, since this task does not have the same actions.

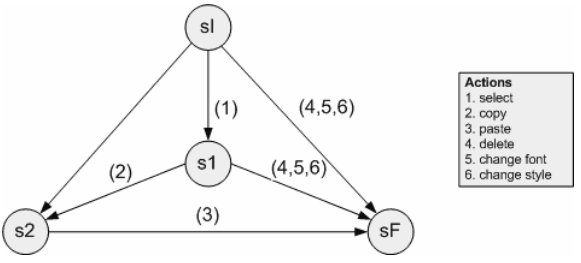


Fig. 1. An example of a state machine for the object *word* of a simple text editor (SI is the initial state and SF is the final state)

Having made the considerations about how to create the state machine, let's turn back to how to use this machine. In our approach, the repetitive pattern detection process deals with two agents: the observer and the pattern detection agents. The basic function of the observer agent is to capture users' action done at the interface. Its implementation, normally, makes use of a common wrapper technique to envelop the functions we want to capture. Its second function is to format the actions done by the user at the interface to be stored in the action history, so that the pattern detector agent could work efficiently. A differentiation of our approach is that the common used action history is divided in tree parts: 1) a list with the *focus defining* actions (the "select" action in the text editor case); 2) a list with the *intermediate* actions, i.e., the actions that do not finalize a task and/or transfer some content to the clipboard (the "copy" action in the text editor case); and 3) a list with the remain actions, i.e., the ones that does not select the focus nor is intermediate. This division is essential to help the detector agent to transmit a complete repetitive task to the generalizer agent. We consider a *task complete* when their actions pass through the state machine, as shown in Fig. 1, from the initial state to the final state. In the second example of section 5 we will show the advantages of this approach. The main problem in this phase is the definition of the action's granularity. In the experiment we made, we decide to capture actions at the level of mouse and menu selection, but this definition is very reliant on the applications for which the agent is been developed.

The basic function of the pattern detector agent is to identify the start and end of a task and, then, look for similar tasks in the task history. In the current PbyD applications, when an action is added to the action history, the detector agent has to verify the possibility to create meaningful patterns with all the existing actions in the action sequences or at least with a limited number of past actions (which composes the current *action context*). Since any action, in this interval, could be the start of a meaningful sequence this involve a great computational effort, as it occurs in Eager [2] an in the APE [12]. When the end of a task is achieved this task will be put in the task history and the generalizer agent will be activated.

In our approach, any time a new action is added to the action history the detector agent will also verify if a task is finished. However, it will not need to compare this action with all actions of the sequence, since only the actions relative to the focus object has some value for the task in question, and these actions have transitions that is determined by the state machine of this object. As a result, any new action could be in one of two situations:

1. It could be interdependent of the last action of the current task, resulting in two possible circumstances: a) the current task is finished, in which case it will be included in the *task history* for the detector agent to look for similar tasks; or b) the current task is not finished, in which case the new action will simply be taking into account as part of the current task and the detection process will continue with the next action.
2. I could not be interdependent of the last action of the current task, what causes the introduction of a *noise* in the action sequence of the current task, resulting in tree possible circumstances:
   - There is already a pending task (a task that has been interrupted by another noise) and the new action is interdependent of the last action of that task. In this case, the new action will be part of the actions of the pending task, and it could finish or not that task. If the pending task is finished, it will be put in the task history and the detector agent should look for repetitive tasks. However, for the new action to be taking into account in the pending task, besides being interdependent on the last action of the pending task, the objects considered in the action as well as in the pending task should be de same (this restriction exists to warrant that the action is a continuation of the pending task);
   - There is already a pending task and the new action is not interdependent of the last action of this task. In this case, no processing will occur, since in the current implementation our pattern detector agent only considers one pending task (it is possible to consider more pending tasks at the same time, but it still not clear if this increase in the implementation complexity is affordable); or
   - There is no pending task. In this case, a new pending task will be created. The current task will be the new pending task and the new action, i.e. the noise, will be the only action of the new current task (the new pending task will be finished afterwards when a new noise arrives in a new task).

3 The Repetitive Task Detection

The pattern detector agent could detect repetitive tasks or sequences of repetitive tasks. In both cases, the tasks or sequences of tasks could be consecutive or not. When identifying repetitions the agent will use two task lists, the *task-sequence* list will be used to store the recently finished tasks that have repetitive tasks in the task history, and the *repetitive-task* list will be used to store the tasks that are already present in the task history and is considered repetitive. Thus, the different types of repetitions possible to be identified by our approach will be detected in the following way:

1. *Consecutive repetitive tasks*: repetitions of this type will be identified after at least three tasks are done, T1, T2 e T3 and necessarily T1 e T2 are similar.

2. *Non-consecutive repetitive tasks*: repetitions of this type will be identified when two repetitive tasks are done not sequentially and a sequence of repetitive tasks is not formed. For example, take the situation in which tasks T1, T2, T3 and T4 are done consecutively, being only tasks T1 and T3 similar ones.
3. *Sequence of consecutive repetitive tasks*: repetitions of this type will be identified after the realization of at least three sequences of tasks. Take the situation in which tasks T1, T2, T3, T4 and T5 are done consecutively, being task T1 similar to T3 and task T2 similar to T4. This way, the sequences [T1,T2] and [T3,T4] will be taking as repetitive.
4. *Sequence of non-consecutive repetitive tasks*: repetitions of this type will be detected when at least two sequences of repetitive tasks are done not sequentially. For example, take the situation in which tasks T1, T2, T3, T4, T5, T6, T7 and T8 are done consecutively and let us consider that there are similarities between tasks T5 and T1, T6 and T2, T7 and T3. Thus, the sequences [T1,T2,T3] are [T5,T6,T7] will be taking as repetitive.

When a repetition detection process is interrupted, the detector agent calls the generalizer agent passing the tasks taking as repetitive in a complete form. If the repetitive tasks do not explicitly indicate the current focus defining action and the intermediate actions (if they exist), the detector agent will take a look at the tree lists that compose the action history to complete the task. For example, if a task with a "copy" action was taken as repetitive the detector agent should look for its focus defining action "select" and its intermediate action "paste" and then pass the whole task to the generalizer. If we do not take into account the tree list in the action history, the detector agent would not able to format the repetitive task in the right way.

It is interesting to notice that, when a repetitive task sequence is detected more that one object type could be involved in the sequence, since more than one task is being taking into account. If this happens, the pattern detector agent should group the tasks with regard to the manipulated objects type and inform the generalizer agent each group individually. This is necessary such that the generalizer agent could consider the details of each type of object. To classify the objects type the pattern detector agent will use domain knowledge by querying the domain application ontology to verify if the object is a subclass of another object. If this is the case, the object type will be the type of its superclass. Otherwise, it will be the type of the task manipulated object itself. The complete detection algorithm could be seen in Fig. 2.

4 An Empirical Analysis of the Pattern Detection Algorithm

To evaluate the quality of our pattern detection agent we have done an empirical analysis with a simple adaptive text editor we develop. As an element of comparison we program a version of the KRM algorithm, used in the APE [12], in Prolog (the language we developed our agent). We compare the algorithms by using the log of the users' actions taking from our text editor, collected for a set of four situations that could happen in daily use of this type of interactive application, being them:

```
Inputs:
   finished-task //most recently finished task
   task-history  //a list of all finished tasks
Local:
  TS //indicate the existence or not of a similar
     //task(s) to the finished task
Begin
For each finished-task do
  TS • false;
  If there exist similar task(s) to the finished task
     in the  task-history then TS • true
  If (TS = true) or (a detection is taken place)then
    If is the beginning of detection then
      Store the detected similar task(s) and the recen
        tely finished task
    elseif TS = true then
      Look for sequences of repetitive tasks
      if there is no such a sequence then
        TS • false   //detection will be interrupted
      endif
  endif
  If TS = false then
    Call the generalizer agent
  endif
endfor
end.
```

Fig. 2. Algorithm for repetitive tasks detection

1. Execution of two repetitive tasks consecutively without noise — where, for example, an user alters the font of two words of the text consecutively, as shown in the log of Fig. 3.

In this situation, we could observe that, by grouping the actions in tasks, our repetition detector agent, using the state machine shown in Fig. 1, will look for patterns

```
1. select(word1, parameterA, parameterB)
2. change-font(word1, parameterC)
3. select(word12, parameterA, parameterB)
4. change-font(word2, parameterC)
```

Fig. 3. Example of consecutive actions in the interface of a text editor

only twice (at the end of action 2 and 4), while the KRM will look for patterns four times, i.e., every time an action is executed.

2. Execution of two repetitive tasks consecutively without noise — where, for example, an user, after having selected and copied a word, *paste* it tree times consecutively in different places, as shown in the log of Fig. 4.

In this situation, our detector agent will look for repetitive patterns only tree times (at the end of actions 3, 4, and 5) while the KRM algorithm would have done this five

```
1. select(word1, parameterA, parameterB)
2. copy(word1)
3. paste(word1, parameterC, parameterD)
4. paste(word1, parameterE, parameterF)
5. paste(word1, parameterG, parameterH)
```

Fig. 4. Example of consecutives actions in the interface of a text editor

times. In addition, with the use of the tree action lists, our detector agent would also take into account the action executed in line 1 and 2 for each repetitive task. In this case, the first repetitive task will have the actions 1, 2 and 4 and the second task will have the actions 1, 2 and 5. Thus, the final repetitive pattern take into account in the generalization process would be complete and different from the pattern detected by KRM algorithm, which will detect as repetitive the action "paste" identified from the actions 4 and 5.

3.Execution of two consecutive repetitive tasks with noise — where, for example, an user copies a word of a text, but before paste that word in the desired place s/he, by mistake, selects another word. Thus, the *copy and paste* task was interrupted by another task, as shown in the log of Fig. 5.

```
1. select(word1, parameterA, paramneterB)      → pending
2. copy(word1)                                 → pending
3. select(word2, parameterC, parameterD)       → noise
4. paste(word1)                          → end of pending task
5. select(word3, parameterA, parameterB)
6. copy(word3)
7. paste(word3))
```

Fig. 5. Example of consecutive actions in the interface of a text editor with noise

In this situation, besides our pattern detector agent being more effective that the KRM algorithm, as we have seen in the previous examples, the use of domain knowledge, in the form of the state machine, allows us to avoid taking action 3 into account in the action sequence, since it is another word (i.e. it is a noise for the task in question). This shows that our agent avoids taking into account noise action in the pattern detection process, what makes the process more efficient.

4. Execution of two repetitive tasks non-consecutively with noise — where, for example, an user copies a word form a text, but before pasting the word in the desired place s/he deletes another word. Thus, the "copy and paste" task was interrupted by another task, as shown in the log of Fig. 6.

In this situation, without the use of domain knowledge, the pattern "select and copy" would be detected. Thus, the generalization that would be inferred and exhibited to the user would be incomplete, since it would not take into account the action "paste." With the use of domain knowledge, at the end of action 3 (that is a noise) the actions 1 and 2 will be put in the *pending task state* and will be waiting the "paste" action to finish the task. With the occurrence of action 5, the pending task will be finished and will be taking into account in the pattern detection process. At the end of action 8, a pattern composed by the actions "select, copy and paste" will be detected.

```
1. select(word1, parameterA, paramneterB)      → pending
2. copy(word1)                                 → pending
3. select(word2, parameterC, parameterD)       → noise
4. delete(word2)
5. paste(word1)                          → end of pending task
6. select(word3, parameterA, parameterB)
7. copy(word3)
8. paste(word3))
```

Fig. 6. Example of non-consecutive actions in the interface of a text editor with noise

Hence, a more coherent generalization will be created an exhibited for the user with our pattern detection agent, what do not happen with the KRM algorithm.

5 Conclusion

The pattern detection processes of current PbyD applications are capable to detecting repetition of action as well as sequences of actions, being them consecutive or not. However, due to the use of simple pattern detection algorithms, like KRM, they are not capable to detect repetition if the actions are not executed exactly in the same order, i.e., they do not detect repletion in the presence of noise.

In this paper, we present an algorithm based on machine learning techniques and the use of domain knowledge to detect repetitive patterns on action sequences done by users in the applications interface. Besides taking into account all the types of repetition that current PbyD applications take, our algorithm also detect repetition in the presence of noise. Nevertheless, we still have the limitation that the actions must be the same, although they do not have to be presented sequentially. This advantage comes from the fact that we take into account the dependencies among actions (which define a task) that could be executed in an object of the domain, i.e., we make use of the application domain knowledge. Another advantage of our approach is the fact that the detector agent could *complete* the tasks (with the help of the tree list of the action history) so that the generalizer agent could make more coherent inferences.

The empirical analyses we executed over a set of action sequences shows that our pattern detection process minimizes the work of the detection agent for almost 50%, when compared with the classic KRM algorithm. But we need to do more tests, with larger sequences and comparing to other algorithms, to make these results more solid. Part of this results could be explained by the use of the notion of tasks associated to an interface object, (i.e., domain knowledge of the application) and by taking into account that the change in the focus object mark the end of a task, allowing our pattern detector agent to activate its detection process only when a task is finished, and not every time an action is executed, as it happens in current implicit PbyD applications.

As future works, we envisage the improvement of our mechanism such that it could detect non-ordered repetitions and repetitions that could have different actions in their tasks. We intend to apply the idea of using a multi-level structured task history proposed in [7] and mentioned in [13]. This will be equivalent to take into account a higher level text element than paragraphs in the text composed by the user interaction section with the application. We think that, in this way, the detector could

take into account higher level tasks in its process and detect patterns in interaction that has equal actions but different order or that has different action and the same order. Consequently, a great number of detection could be done and a great number of generalizations could be inferred to automate the users' tasks. Another point to study is to take into account more than one pending task, what could improve the quality of the noise treatment by the detector agent.

References

1. Cypher, A. (ed) Watch What I Do: Programming by Demonstration. The MIT Press, Cambridge, MA, 1993.
2. Cypher, A Eager: Programming repetitive tasks by demonstration. In Cypher, A. (ed) Watch What I Do: Programming by Demonstration. The MIT Press, Cambridge, MA, 1993, pg. 205-217.
3. De Souza, C. S. The Semiotic Engineering of Human-Computer Interaction. MIT Press, Cambridge, MA, 2005.
4. Godoi, M. S. Integração de Técnicas de Programação por Demonstração e Geração Automática de Interfaces. Trabalho de Graduação. Departamento de Informática, UEM, 2004..
5. Kammersgaard, J. Four different perspectives on human- computer interaction. In International Journal of Man- Machine Studies. 1988, No. 28, p.343-362.
6. Karp, R. M; Miller, R. E and Rosenberg, A. L. Rapid Identification of Repeated Patterns in Strings, Trees and Arrays. In 4th Annual ACM Symposium on Theory of Computing, Denver, Colorado, 1-3 May, 1972, pg. 125-136.
7. Kosbie, D. S.; Myers, B. A. A System-Wide Macro Facility Based on Aggregate Events: A Proposal. In: Cypher, A (ed) Watch What I Do: Programming by Demonstration. Cambridge: MIT Press, 1993. 433-444.
8. Lau, T., Wolfman, S. A., Domingos, P., Weld, D. S. Learning Repetitive Text-Editing Procedures with SMARTedit. In: Lierberman, H: Your Wish Is My Command: Programming by Example. Morgan Kaufmann, Sao Francisco CA, 2001, p. 209-225.
9. Levinson, S. C. Pragmatics. Cambridge University Press, Cambridge, UK. 1983.
10. Norman, D. A. How might people interact with agents. In: *CACM*, 37(7), 1994, 68-71.
11. Penteado, R. R. de M. O Uso de Conhecimento Prévio do Domínio em Aplicações na Programação por Demonstração. Dissertação de Mestrado. Departamento de Informática, UEM, 2004.
12. Ruvini, J. and Dony, C. Learning Users' Habits to Automate Repetitive Tasks. In: Lierberman, H Your Wish Is My Command: Programming by Example. Morgan Kaufmann, Sao Francisco CA, 2001, p. 271-295.
13. Ruvini, J. D. The Challenges of Implicit Programming by Example. In: IUI'04, 2004, Madeira, Funchal, Portugal, 13-16.

Supporting Ontology-Based Semantic Matching of Web Services in MoviLog*

Cristian Mateos, Marco Crasso, Alejandro Zunino, and Marcelo Campo

ISISTAN Research Institute, UNICEN University - Campus Universitario,
(B7001BBO) Tandil, Bs. As., Argentina
cmateos@exa.unicen.edu.ar

Abstract. The Web is moving towards the creation of a worldwide network of Web Services known as the Semantic Web. This new environment will allow agents to autonomously interact with Web information and services. This paper presents Apollo, an infrastructure which offers semantic matching and discovery capabilities to MoviLog, a platform for building mobile agents on the Semantic Web. Examples and experimental results showing the practical usefulness of Apollo are also reported.

1 Introduction

Once a big repository of pages and images, the Web is evolving into a worldwide network of *Web Services* called Semantic Web [1]. A Web Service [2] is a distributed piece of functionality that can be published, located and accessed through standard Web protocols. The goal is to achieve automatic interoperability between applications by means of an infrastructure to use Web resources.

Several researchers agree that mobile agents will have a fundamental role in the materialization of this vision [3]. A mobile agent is a program able to migrate within a network to perform tasks or interact with resources. Mobile agents have suitable properties for exploiting the potential of the Web, such as support for disconnected operations, heterogeneous systems integration and scalability [4].

Despite these advantages, many challenges remain in order to glue mobile agents with Web Services technology. Most of them are a consequence of the nature of the WWW, since from its beginnings Web content has been designed for human interpretation [5]. Hence, unless content is described in a computer-understandable way, mobile agents cannot autonomously take advantage of the capabilities of Web resources, thus forcing developers to write hand-coded solutions that are difficult to reuse and maintain. This fact, together with the inherent complexity of mobile code programming, has affected the massive adoption of mobile agents and limited its use to small applications.

Indeed, there is a need for an agent infrastructure that addresses these problems and preserve the key benefits of mobile agent technology for building distributed applications. To cope with this, a platform for building logic-based

* This research has been partially supported by CONICET and ANPCyT (PICT 11-13465).

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 390–399, 2006.

mobile agents on the WWW named MoviLog [6] has been developed. MoviLog encourages the usage of mobile agents by supporting a mobility mechanism named RMF (Reactive Mobility by Failure). RMF allows programmers to easily code and deploy mobile agents on the Semantic Web without worrying about Web Services location or access details. Furthermore, to consider the semantics of services, it provides an infrastructure for semantic matching and discovery of Web Services named Apollo, which allows for a truly automatic interoperation between mobile agents and services with little development effort.

The next Sect. introduces semantic Web Services. Section 3 presents MoviLog. Section 4 describes Apollo. Then, Sect. 5 explains an example. Experimental results are reported in Sect. 6. Section 7 surveys the most relevant related work. Finally, Sect. 8 draws conclusions.

2 Semantic Web Services

Web Services are a suitable model to allow systematic interactions of applications across the WWW. The Web Services model relies on XML, a structured language that formalizes HTML. In this sense, SOAP[1], a communication protocol based on XML, has been developed. In addition, languages for describing Web Services have been proposed. The most notorious example is WSDL[2], an XML-based language for describing services as a set of operations over SOAP messages. From a WSDL specification, a program can determine the specific services a Web site provides, and how to use and invoke these services.

UDDI[3] defines mechanisms for searching and publishing services written in WSDL. Providers register information about the services they offer, thus making it available to clients. The information managed by UDDI ranges from WSDL files describing services interfaces to data for contacting providers.

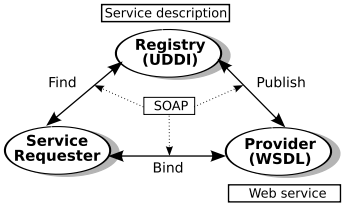


Fig. 1. Web Service conceptual architecture

The most widely accepted architecture for Web Services is depicted in Fig. 1. A Web Service is defined as an WSDL interface describing a set of Web-accessible operations. The provider creates a WSDL document describing their Web Service and publishes it to a UDDI registry. A requester can browse registries to find a service that matches their needs along with the corresponding WSDL document. Finally, the requester can bind to the provider by invoking any of the operations defined in the WSDL document.

The weakest point of the architecture is that it does not consider the semantics of services. To achieve a truly automatic interaction between agents and Web Services, each service must be described in a computer-understandable way. Some languages for service metadata annotation have emerged, such as

[1] SOAP Specification: http://www.w3.org/TR/soap/
[2] WSDL Specification: http://www.w3.org/TR/wsdl/
[3] UDDI: http://www.uddi.org

RDF[4] and OWL [7], whose goal is to provide a formal model for describing the concepts involved in services. In this way, agents can autonomously understand and reason about the precise functionality a Web Service performs, thus leading to a complete automatization of Web applications.

3 MoviLog

MoviLog [6] is a platform for programming mobile agents in the WWW. The execution units of MoviLog are Prolog-based strong mobile agents named *Brainlets*. Besides providing basic mobility primitives, the most interesting aspect of MoviLog is the notion of *reactive mobility* [8]. RMF is a mobility model that reduces agent developing effort by automating decisions such as when or where to migrate upon a *failure*. Conceptually, a failure is defined as the impossibility of an executing agent to obtain some required resource at the current site.

Roughly, each Brainlet possess Prolog code that is organized in two sections: *predicates* and *protocols*. The first section defines the agent behavior and data. The second section declares rules that are used by RMF for managing mobility. RMF states that when a predicate declared in the protocols section fails, MoviLog moves the Brainlet along with its execution state to another site that contains definitions for the predicate. Indeed, not all failures trigger mobility. The idea is that normal predicates are evaluated with the regular Prolog semantics, but predicates for which a protocol exists are treated by RMF so their failure may cause migration. The next example presents a simple Brainlet whose goal is to solve an SQL query given by the user on a certain database:

```
PROTOCOLS
 protocol(dataBase, [name(N), user(U), password(P)]).
CLAUSES
 doQuery(DBName, Query, Res):-
  ParamList=[name(DBName), user('default'), password('')],
  dataBase(ParamList, DBProxy),
  doQuery(DBProxy, Query, Res).
?-sqlQuery(DBName, Query, Res):- doQuery(DBName, Query, Res).
```

PROTOCOLS section declares a protocol stating that the evaluation of *dataBase(...)* predicate must be handled by RMF. In other words, the RMF mechanism will act whenever an attempt of accessing a certain database fails in the current site. As a result, RMF will transfer the agent to a site containing a database named *DBName*. After connecting to the database, the Brainlet will execute the query, and then return to its origin. Note that the protocol does not specify any particular value to the properties of the requested resource (i.e. N, U and P variables), which means that all unsuccessful attempts to access locally *any* database with *any* username-password will trigger reactive mobility.

Despite the advantages RMF has shown, it is not adequate for developing Web-enabled applications because it lacks support for interacting with Web

[4] RDF Specification: http://www.w3.org/RDF/

resources. To overcome this limitation, RMF has been adapted to provide a tight integration with Web Services [9]. Also, in order to take advantage of services semantics, an infrastructure for managing and reasoning about Web Services metadata named Apollo has been built. The rest of the paper focuses on Apollo.

4 Semantic Matching in MoviLog

Ontologies are used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms [1]. In order to infer knowledge from ontologies, Apollo includes a Prolog-based reasoner implemented as a set of rules to determine semantic similarity between any pair of concepts.

4.1 Representing Ontologies in Prolog

The reasoner is built on top of the OWL-Lite language [7]. Interestingly, OWL-Lite is easily translatable to Prolog, since it has a Description Logic equivalent semantics, which are a decidable fragment of first-order logic [10].

Basic OWL-Lite constructors for classes and properties are represented as simple facts, while higher-level relationships are expressed as RDF *triples,* a structure *triple(subject,property,value)* stating that *subject* is related by *property* to *value.* OWL-Lite features such as cardinality, range and domain constraints over properties are also translated into triple. For example, *triple(author, range, person)* states that *author* must be an instance of class *person.* OWL-Lite equality, inequality and transitive sentences are also supported. For example, if *author* and *writer* were equivalent properties, then *triple(article,writer,person)* holds.

Fig. 2 depicts a simple ontology for documents.

The ontology defines that both *thesis* and an *article* are *documents* having at least one *author.* A thesis has also an *advisor.* Both *author* and *advisor* are properties within range *person.* A document is composed of a *title*, a *language* and *sections.* Finally, every section has *content.* In the previous rules two new concepts appear: *Thing* and *owl:string.* Thing is the parent class of all OWL classes. Also, OWL includes some built-in datatypes such as *owl:string*, *owl:long*, *owl:boolean*, to name a few, which allow to define literal properties.

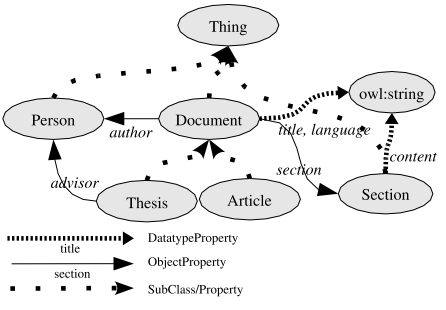


Fig. 2. An ontology for documents

4.2 Matching Concepts

Ontologies are used to describe data and services in a machine-understandable way. In automated data migration systems, developers use ontologies to annotate

semantically their data structures. A process may then be executed to migrate a record from a source database to a sufficiently similar record in a target database. In automated Web Services discovery systems, agents usually try to find a enough similar service to accomplish their current goal. The problem is indeed to define what "enough similar" means.

The degree of match between two concepts depends on their distance in a *taxonomy tree*. A taxonomy may refer to either a hierarchical classification of things, or the principles underlying the classification. Almost anything (objects, places, events, etc.) can be classified according to some taxonomic scheme. Mathematically, a taxonomy is a tree-like structure that categorizes a given set of objects. Like [11], Apollo defines four degrees of similarity between two concepts X and Y:

- **exact** if X and Y are individuals belonging to the same or equivalent classes.
- **subsumes** if X is a subclass of Y (for example *thesis* and *document*).
- **plug-in** if Y is a subclass of X (for example, *document* and *thesis*).
- **fail** occurs when none of the previous labels could be stated.

As shown in Fig. 3, this scheme is enhanced by taking into account the distance between any pair of concepts in a taxonomy tree. It can be clearly stated that *c2* is more similar to *b1* than to *a1*, and their similarity is labeled as **plug-in**.

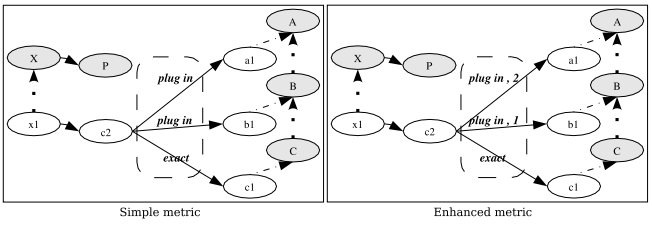


Fig. 3. Enhanced degree of match

The matchmaking algorithm consists of Prolog rules for measuring the taxonomic distance between concepts. The rule *match(C0, C1, L, D)* returns the distance D between concept C0 and concept C1 under label L. Distance between a class and its direct related superclasses is defined as one. Distance for indirect subclassing is defined recursively. For example, computing distance for C0=article and L=subsumes produces {C0=document, D=1} and {C1=thing, D=2}. Matchmaking support for properties works in a similar way.

4.3 Semantic Web Services Discovery

In order to perform a semantic search of a Web Service instead of a less effective keyword-based search (i.e. by service name), an agent needs computer-interpretable descriptions about the functionality of services. Ontologies can be used for representing such descriptions. OWL-S [12] is a worldwide effort which aims at creating a standard service ontology, describing services and how they must be

invoked, but not how to semantically find them. Service Profile is an OWL-S sub-ontology which offers support for semantic description of services functionality, arguments, preconditions and effects. The current prototype of Apollo combines Service Profile and UDDI to build a semantic Web Services discovery system.

Apollo allows a Web Service provider to publish its services to pre-configured UDDI registries, and to annotate these services by using concepts from a shared ontology database. Those annotations, which are made in terms of properties from the Service Profile ontology, are stored onto a *Semantic Repository*. Finally, the semantic annotations and UDDI information (businessService and tModels) for a service are associated through a UDDI service key.

Apollo permits applications to search for Web Services both in UDDI nodes and in its Semantic Repository. A search request may contain a semantic and a syntactic search condition, which are forwarded to the semantic reasoner and UDDI, respectively. The former is a collection of *<property, expected_value>* pairs, each one describing the desired conceptual value for some specific relationship *property* within the Service Profile ontology. For example, *<owls:hasOutput, thesis>* ask for services whose output is semantically similar to a thesis concept. A syntactic search condition is equivalent to a UDDI-like search request.

5 A Sample Scenario

Suppose a network comprising sites which accept Brainlets for execution. Some of these sites offers Web Services for translating different kind of documents to a target language. Every time a client wishes to translate a document, an agent is asked to find the service that best adapts to the kind of document being processed. In order to add semantics features to the model, all sites publish and search for Web Services by using Apollo, and services are annotated with concepts from the ontology presented in Sect. 4.1.

It is assumed the existence of different instances of Web Services for handling the translation of a specific kind of document. For example, translating a plain document may differ from translating a thesis, since a more smart translation can be done in this latter case: a service can take advantage of a thesis' keywords to perform a context-aware translation. Nevertheless, note that a thesis could be also translated by a Web Service which expects a Document concept as an input argument, since Thesis concept specializes Document according to our ontology.

When a Brainlet gets a new document for translation, it prepares a semantic query. In this case, the agent needs to translate a thesis to English. Before sending the service query, the Brainlet sets the service desired output as a Thesis. Also, the Brainlet sets the target language as English and the source document kind as Thesis, and then the semantic search process begins. Apollo uses semantic matching capabilities to find all existing Translation services. Suppose two services are obtained: a service for translating theses (*s1*) and a second service (*s2*) for translating any document.

After finding a proper list of translation Web Services, Apollo sorts this list according to the degree of match computed between the semantic query and

services descriptions. In the example, the degree of match for *s1* is greater than for *s2*, because *s1* outputs a Thesis (***exact*** matchmaking) while *s2* matchmaking was labeled as ***subsumes*** with distance one.

```
PROTOCOLS
 protocol(service,[name(translate),in([thesis,english]),
                   out(thesis)]).
CLAUSES
 thesis([title('Title'), author('Author'), language(spanish),
         advisor('Advisor'), sections([...])}).
 ?-translate(TargetLang, Result):-
  Props=[name(translate),in([thesis,TargetLang]),out(thesis)]
  service(Props,WSProxy), thesis(Th),
  executeService(WSProxy, [Th, TargetLang], Result).
```

The previous code implements the Brainlet discussed so far. When the *webService(...)* predicate is executed, RMF contacts Apollo in order to find candidate services that semantically match the Brainlet's request. The evaluation of the predicate returns a proxy which is used to effectively access the service. The way the service is contacted (migrate to the service location or remotely invoke it) depends on access policies based on current execution conditions (network load, agent size, etc.) managed by the underlying platform.

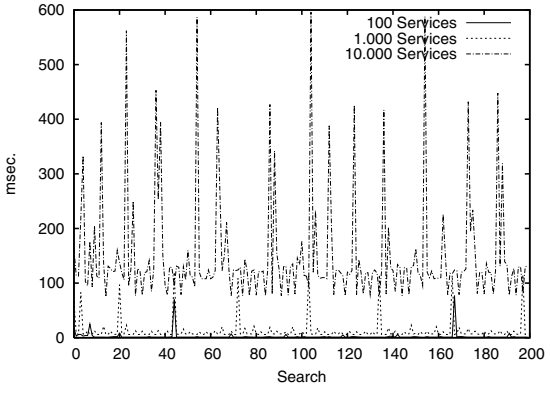
To sum up, the Brainlet has obtained a Web Service for execution using data semantic information rather than a syntactic description. To imagine a nonsemantic matching scenario, assume that is defined a syntactical categorization of services for translating documents. Such a categorization will typically have a tree-like structure with a root node labeled *"*Document translator*"*. The root will have two children nodes labeled *"*Article Translator*"* and *"*Thesis Translator*"*, respectively. Without a semantic description about the kind of document each service is able to translate, the only way to find proper services is by their name, a pure syntactic and rigid mechanism. In this way, the logic to determine which service is appropriate for translating each kind of document remains hardcoded in the agent. Also, when a new kind of document unknown to the agent is added, its implementation becomes obsolete.

6 Experimental Results

6.1 Performance Tests

Test cases were conducted with regard to various size of the description database. Both Apollo and all test applications were deployed on an Intel Pentium 4 working at 2.26 GHz and 512 MB of RAM, running Sun JVM 1.4.2 on Linux.

The database was automatically created based on two ontologies: stock management and car selling. Each service description was composed of five properties: input, output, category, preconditions and effects. For example, the concepts involved in a service providing a quote for a sport car are *cs:sportcar* (input), *cs:quote* (output) and *cs:car_quoting* (functionality). Searches were simulated by using randomly-generated conditions and expected results.



# of services	10000	1000	100
ART (msec.)	2.37	12.65	149.33

Fig. 4. Searches vs. response time

Figure 4 shows the relation between database size and the time for processing 200 different searches by test case. The worst response time is 600 ms., even with 10.000 Web Services. It is worth noting that the peaks of the curves are a consequence of overhead introduced by the JAVA garbage collector. Similarly, the table summarizes the resulting average response (ART) time for 600 random searches. It shows that Apollo performance was outstanding. The ART was less than 200 ms., even with 10.000 services stored in the database.

6.2 Comparison Against ASP

This subsection presents a brief comparison between the implementation of our reasoner and another implementation using ASP [13,14]. ASP is a logical programming language with a syntax similar to Prolog but, instead of a query driven interface, ASP offers a *result set* driven interface. A result set (or stable model) is a view of all rules that can simultaneously coexist. Given an ASP program and a query, the output is a stable model. ASP has been recently proposed as a formalism for providing inference capabilities for the Semantic Web.

Comparison between the two implementations showed that a result set driven interface is less adequate for semantic matchmaking, since it requires to analyze the result set in order to find the rule(s) storing the resulting semantic similarity. Notice that the output of any ASP program is usually a big list of those rules which comforms a stable model for a query where contradictory knowledge does not exist. This clearly adds an overhead when parsing the output of a query.

Both implementations represent OWL-Lite information as RDF triples, which is the only feature they have in common. Rules for computing taxonomic distance from Apollo's implementation are more concise and clear than ASP's. On the other hand, the performance of Apollo's reasoner is benefited from the query driven interface inherent to Prolog. The ASP reasoner produced large result sets even for simple queries and a description database of 100-1000 services (39 predicates against 1 for the case of Prolog).

7 Related Work

There are some proposals for semantic matching, publication and discovery of Web services [15,16]. One major limitation of these approaches is that their matching scheme do not take into account the distance between concepts within a taxonomy tree. As a result, similarity related to different specializations of the same concept are wrongfully computed as being equal.

The most relevant work to our approach is the OWL-S Matchmaker [12], a semantic Web Service discovery and publication system that is UDDI-compliant. It includes a semantic matching algorithm that is based on service functionality, and data transformation descriptions which are made in terms of service input and output arguments. Service search requests are enriched with concepts describing the list of services that match a required data transformation. This approach does not support taxonomic distance between concepts either.

With respect to the plethora of work on mobile agents tools for the Semantic Web, some interesting advances are ConGolog [17], IG-JADE-PKSLib [18] and MWS [19]. However, these approaches present the following problems: bad performance/scalability (IG-JADE-PKSLib), no/limited mobility (IG-JADE-PKSLib, ConGolog) and lack of support for common agent requirements such as knowledge representation and reasoning (MWS). Also, none of the previous platforms supports semantic matching and discovery of Web Services.

Finally, some modeling frameworks for creating Semantic Web enabled Web Services have been recently proposed, such as SWSA [20] and WSMO[5]. However, rather than specific components, these models offer abstract specifications for materializing infrastructure and ontologies for the Semantic Web.

8 Conclusion and Future Work

This paper introduced Apollo, an infrastructure which offers semantic matching and discovery capabilities, and described how this support has been integrated to a mobile agent platform named MoviLog.

Unlike previous work, Apollo defines a more precise matching algorithm, implemented on top of a Prolog reasoner which offers inference capabilities over OWL-Lite annotations. In addition, the integration of MoviLog with this support enables the development of mobile agents that interact with Web-accessible functionality published across the WWW. This leads to the creation of an environment where every site can publish its capabilities as Semantic Web Services to which Brainlets can find and access in a fully autonomous way.

Regarding Apollo, some issues remain to be solved. First, OWL Lite needs to be replaced by a more powerful and expressive language (e.g. OWL DL or OWL Full). Second, Ontologies Database content must be enhanced in order to provide a framework to describe, publish and discover any type of semantically annotated Web resource (pages, blogs, other agents), and not just Web Services.

[5] WSMO: http://www.wsmo.org

Thereby a software agent would be able to autonomously interact with any type of Web content defined in a machine-processable way.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic Web. Scientific American **284** (2001) 34–43
2. Vaughan-Nichols, S.: Web services: Beyond the hype. Computer **35** (2002)
3. Hendler, J.: Agents and the semantic Web. IEEE Intelligent Systems **16** (2001) 30–36
4. Lange, D.B., Oshima, M.: Seven good reasons for mobile agents. Communications of the ACM **42** (1999) 88–89
5. McIlraith, S., Son, T.C., Zeng, H.: Semantic Web services. IEEE Intelligent Systems, Special Issue on the Semantic Web **16** (2001) 46–53
6. Zunino, A., Campo, M., Mateos, C.: Simplifying mobile agent development through reactive mobility by failure. In: SBIA'02. Volume 2507 of Lecture Notes in Computer Science., Springer-Verlag (2002) 163–174
7. Antoniou, G., van Harmelen, F.: Web ontology language: OWL. In Staab, S., Studer, R., eds.: Handbook on Ontologies in Information Systems, Springer-Verlag (2003)
8. Zunino, A., Mateos, C., Campo, M.: Reactive mobility by failure: When fail means move. Information Systems Frontiers. Special Issue on Mobile Computing and Communications **7** (2005) 141–154 ISSN 1387-3326.
9. Mateos, C., Zunino, A., Campo, M.: Extending movilog for supporting Web services. Computer Languages, Systems & Structures (2006) To appear.
10. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The description logic handbook: theory, implementation, and applications. Cambridge University Press (2003)
11. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic matching of Web services capabilities. In: ISWC'02. Volume 2342., Springer-Verlag (2002)
12. Paolucci, M., Sycara, K.: Autonomous semantic Web services. IEEE Internet Computing **7** (2003) 34–41
13. Marchi, M., Mileo, A., Provetti, A.: Declarative policies for Web service selection. In: POLICY'05, Sweden, IEEE Computer Society (2005) 239–242
14. Ruffolo, M., Leone, N., Manna, M., Saccà, D., Zavatto, A.: Exploiting ASP for semantic information extraction. In: ASP'05. Volume 142 of CEUR Workshop Proceedings. (2005)
15. Horrocks, I., Patel-Schneider, P.F.: A proposal for an OWL rules language. In: WWW'04, ACM Press (2004) 723–731
16. Chiat, L.C., Huang, L., Xie, J.: Matchmaking for semantic Web services. In: SCC'04, IEEE Computer Society (2004) 455–458
17. McIlraith, S.A., Son, T.C.: Adapting Golog for programming the semantic Web. In: Common sense'01. (2001)
18. Martínez, E., Lespérance, Y.: IG-JADE-PKSlib: An agent-based framework for advanced web service composition and provisioning. In: WSABE'04. (2004) 2–10
19. Ishikawa, F., Yoshioka, N., Tahara, Y., Honiden, S.: Towards synthesis of web services and mobile agents. In: WSABE'04. (2004)
20. Burstein, M., Bussler, C., Zaremba, M., Finin, T., Huhns, M.N., Paolucci, M., Sheth, A.P., Williams, S.: A semantic Web services architecture. IEEE Internet Computing **9** (2005) 72–81

Learning Similarity Metrics from Case Solution Similarity*

Carlos Morell, Rafael Bello, Ricardo Grau, and Yanet Rodríguez

Computer Sciences Department.
Universidad Central de Las Villas. Cuba
{cmorellp, rbellop, rgrau, yrsarabia}@uclv.edu.cu

Abstract. Defining similarity metrics is one of the most important tasks when developing Case Based Reasoning (CBR) systems. The performance of the system heavily depends on the correct definition of its similarity metric. To reduce this sensitivity, similarity functions are parameterized with weights for features. Most approaches to learning feature weights assume CBR systems for classification tasks. In this paper we propose the use of similarity between case solutions as a heuristic to estimate similarity between case descriptions. This estimation is used to adjust weights for features. We present an experiment in the domain of Case Based Process Planning that shows the effectiveness of this approach.

1 Introduction

The notion of similarity or distance in the feature space that describes the observations is the basis of several pattern recognition and machine learning methods. In pattern recognition the nearest neighbor methods [1] are examples of similarity based methods, in artificial intelligence the instance based learning [2] and case based reasoning [3] methods work under the assumption that "similar problems have similar solutions". Case-based reasoning has become a popular technology for building knowledge-based Systems. The CBR problem solving process consists of four phases called Retrieve, Reuse, Revise and Retain that are known as the R^4 cycle [3].

The basic architecture of a CBR system consists of a case base, a retrieval procedure for finding similar cases and an adaptation method used to fit solutions to the new requirements. The case base is a collection of solved problems in the application domain. The retrieval procedure seeks the case base to find the most similar case or cases to the problem being solved. Once that case has been found, its solution is used as a basis to build the solution of the new problem. The adaptation method is used for this last task.

The relationship between the retrieval procedure and the adaptation method is crucial because it is desirable to recover cases requiring smaller adaptation efforts. It is convenient to adjust the similarity criteria to minimize the amount of transformations to be carried out over the retrieved solutions.

* This work was supported in part by VLIR (Vlaamse InterUniversitaire Raad, Flemish Interuniversity Council, Belgium) under the IUC Program VLIR-UCLV.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 400 – 408, 2006.

Most approaches to learning feature weights assume CBR systems for classification tasks [4]. In this work we present a method to adjust feature weights in such a way that the resulting similarity function allows the recovering of cases whose solutions require fewer transformations. Specifically, we apply this method in the domain of case based process planning. Planning by adapting previous successful plans is an attractive reasoning paradigm because it decreases the computational complexity [5].

2 Basis

2.1 Case-Based Reasoning

In CBR, a new problem is solved by finding a similar past case, and reusing it in the new problem situation. The main tasks that any CBR methods have to deal with are: the identification of the current problem situation, the search for a previous case similar to the new one, the utilization of that case to suggest a solution to the current problem, the evaluation of the proposed solution, and finally the system updates by learning from this experience. The CBR paradigm comprises a range of different methods for organizing, retrieving, utilizing and indexing the knowledge retained in past cases. A case usually denotes a problem situation, a previously experienced situation which has been captured and learned in such a way that can be later reused for solving future problems. Cases may be kept as concrete experiences or a set of similar cases can make up a generalized case. An initial description of a problem defines a new case. This new case is used to retrieve a case from a collection of previous cases; the retrieved case is combined with the new case into a solved case (a proposed solution to the initial problem).

Adaptation is the process of fixing up an old solution to meet the new problem. In most occasions, an adaptation process is absolutely necessary because retrieved cases never match exactly the current problem. Adaptation is treated differently in many case-based systems. Some systems avoid using adaptation while other systems use general adaptation methods such as derivational and transformational adaptation [6]; most systems use domain-specific adaptation procedures. In planning the adaptation process is always required because solution space is usually huge. It is extremely unlikely that a plan could be reused for a new problem without modification [5].

Case adaptation takes place within a larger context, including the interaction with other components of the case-based system. One way to ease the problem is to tie other components more closely into the adaptation process. Especially important is the relation with the retrieval procedure because the difficulty of case adaptation depends on the cases being retrieved. Improvements on retrieval can significantly improve the adaptation task [7]; besides, the flexibility of case-based systems depends on their retrieval and adaptation processes [8]. The link between these two modules has been a research topic addressed by several authors [9] and [10].

Other works related to this subject are the following: In [11], the adjustment of the similarity criteria upon the adaptation process is analyzed. The elaboration of a similarity measure based on an adaptability estimation guided by the elicitation of discrepancy reduction is stated in [12] among the main tasks in case-based reasoning applications. In DEJAVU [13], adaptability is used to guide retrieval. In KRITIK [14]

candidate cases that tend to have more easily adaptable matches are preferred. M. Brand et al. [15] and S. Fox and D Leake [16] index cases based on their adaptability. In [17], the retrieval procedure in case-based planning is incrementally improved by detecting and explaining failures that might occur.

2.2 Case Representation

A case c is a tuple <d, s> ∈ D x S consisting in a problem's description and an associated solution. A case description will be represented using a feature-value representation while case solution can be arbitrary complex. Case base is assumed to be a collection CB={<d_1,s_1>,<d_2,s_2>,…,<d_n,s_n>}of precedent cases. The similarity $_d$ between two object descriptions d and d_i is computed by the expression (1) where w_a is the weight for feature a and sim_a is the local similarity measure for feature a. In general, it is assumed that $\sum w_a$=1 and $sim_a \in [0,1]$ for all attributes, yielding the similarity between two descriptions $_d(d,d_i) \in [0,1]$.

$$\sigma_d(d,d_i) = \sum_{a=1}^{n} w_a \cdot sim_a(d,d_i) \tag{1}$$

The similarity $_S(s,s_n)$ between two object solutions is an a posteriori criterion. When solving new problems the solution part of the query object is usually not available. The similarity measure between descriptions is used to approximate some form of *utility*. This means, the larger the similarity between the query and the case the more useful is the case for the query. As discussed in [18] the main purpose of the retrieval step in CBR is to select cases whose solutions can be adapted to the new problem. Hence, the retrieval should certainly select the case that is most suitable for adaptation. Even if no adaptation is to be performed by the CBR system, i.e., solution reuse is left to the user, the retrieval should select re-useable cases.

3 Calculating Feature Weights

Given a new problem with description p, the expression $_D(p,d_i)$ quantifies the similarity between this problem and the *i-th* case of the *CB* with description d_i. This value is "a priori", a measurement of the utility of the solution s_i to solve the new problem.

In addition for each known case $q \in CB$, the function σ_S generates an order of the CB based on the utility of the solution of each case with respect to the solution s_q of case q:

$$O_{\sigma_s}(q) = \{c_1, c_2, \ldots, c_n\} \text{ such that } \forall i\, \sigma_S(s_q, s_{C_i}) \leq \sigma_S(s_q, s_{C_{i+1}})$$

This order is considered as "the absolute or valid truth for case q" because it responds to the order of utility of each case of the CB with respect to the solution of problem q .Obviously, at least the first case according to this order determines the solution for q.

The function σ_D generates an order in analogous way, but not necessarily with identical results, perhaps altered by a permutationπ, this is:

$O_{\sigma_D}(q) = \{c_{\pi_1}, c_{\pi_2}, \ldots, c_{\pi_n}\}$ such that $\forall i\ \sigma_D(d_q, d_{C_{\pi_i}}) \leq \sigma_D(d_q, d_{C_{\pi_{i+1}}})$

The function σ_D is "imperfect", and therefore "perfectible". A measurement of the imperfection of this order for a known case q can be measured by the following expression that we will name "disorder" generated by σ_D.

$$D_{\sigma_D}(q) = \frac{\sum_{i=1}^{n} |i - \pi_i|}{\left\lfloor \frac{n^2}{2} \right\rfloor} \tag{2}$$

The integer part of $n^2/2$, which appears at the denominator, is the maximal number of possible permutations and is used for normalization purposes.

The problem consists in finding a similarity function between descriptions (σ_D) so that the disorder $D_{\sigma_D}(q)\ \forall q \in CB$ was minimized. When a similarity function like expression (1) is used, there are two parameters to modify with the previous purpose: the vector of weights w and the local similarity functions. It is preferable to work with the set of weights by its flexibility, which leads to the formulation of the following problem:

Problem: To find the weight vector w so that, given a known local similarity functions, the resulting global similarity function fulfills that $D_{\sigma_D}(q)\ \forall q \in CB$ is minimum.

The solution to the previously stated problem is to minimize the disorder caused by the similarity function over the entire case base. It is equivalent to minimize the following error function:

$$e_w(q) = \sum_{i}^{n} (\sigma_D(d, d_i) - \sigma_S(s, s_i))^2 \quad \forall q \in CB \tag{3}$$

The proposal is that a minimization of error e_w(q) for all q $\in$ *CB*, by means of feature weighting, will lead to a similarity function that minimize the disorder $D_{\sigma_D}(q)\ \forall q \in CB$.

The rationale of this proposal is straightforward: minimization of e_w (q), for all the $q \in CB$, particularly implies the minimization of the difference according to σ_D and σ_S between the separation of a well-known case q with respect to the general CB. It is therefore presumable that, in conditions of minimum, the order determined by σ_D is similar enough to the order determined by σ_S, that is, there must exists a tendency to the identity permutation. This would be valid for all cases $q \in CB$ although the result was different.

The Bayesian induction consists of applying this principle to a new problem p with known description and unknown solution. The inductive bias can be considered from the cases of success and error with data of the BC, preferably not used during the learning.

The average quadratic error is calculated according to the expression (4):

$$E(w) = \frac{1}{n^2} \sum_{j=1}^{n} e_w(c_j) = \frac{1}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} \left(\sigma_S^w(c_j, c_k) - \sigma_D(c_j, c_k)\right)^2 \tag{4}$$

A formulation of the error first derivative with respect to the weights is the following one:

$$\frac{\partial E(w)}{\partial w_i} = \frac{2}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} \left(\sigma_S^w(c_j, c_k) - \sigma_D(c_j, c_k) \right) \cdot \frac{\partial \sigma_S^w(c_j, c_k)}{\partial w_i} \tag{5}$$

Expression (5) is equivalent to the following expression:

$$\frac{\partial E(w)}{\partial w_i} = \frac{2}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} \left(\sigma_S^w(c_j, c_k) - \sigma_D(c_j, c_k) \right) \cdot sim_i(c_j, c_k) \tag{6}$$

And now it is feasible to use the conjugated gradient method to minimize the error modifying the weights in the direction of the Conjugated Gradient.

Finally, the description of the algorithm to find an optimal set of weights is as follows:

```
1. initialize weight-vector w
2. determine for each case from the CB its utility  s with re-
   spect to the rest
3. compute the error E(w) according to (4)
4. initialize the learning rate
5. repeat until  becomes very small
```

a. Generate a new function $\sigma_D^{w'}(q,c)$ so that:

$$w'_i = w_i - \frac{\partial E(w)}{\partial w_i} \cdot \lambda$$

b. normalize the new weights

c. compute E(w') according to (4)

d. **if** E(w') < E(w)
then $w = w'$
else = /2

The conjugated gradient method is a search algorithm that tries to optimize a system with respect to an error function. This optimization is accomplished by an iterative search for a local minimum of the error function. First, the weight vector w is initialized to random values. After the calculation of σ_S between every pair of cases, the error E(w) is determined. Finally, the algorithm does a number of learning steps modifying the weights in the direction of a local minimum. If the vicinity of the minimum has the shape of a long, narrow valley, the minimum is reached in far few steps.

4 Empirical Evaluation and Results

The original purpose of this method was the development of a Computer Aided Process Planning System for symmetrical-rotational parts. The selection of the appropriate weights for each feature becomes a very difficult task for the experts in the

application domain basically due to its lack of knowledge about the process of adaptation. They can decide the importance of each feature from the syntactic point of view but once a technique of adaptation is used, these approaches fail at all.

When the retrieval method used is based on similarity functions like the one of the expression (1), weights can be a decisive factor in the utility of the recovered cases. For that reason, a problem of interest in the cases-based planning is to determine the importance of the features a_i so that cases whose plans might require a smaller adaptation effort are recovered. In the previous section, a method for feature weighting is outlined, applying an iterative procedure according to the criterion of minimizing the effort of plan adaptation.

4.1 Application to the Domain of Case-Based Process Planning

In this particular application, the effort can be measured as the amount of T-*operators* that are needed, by using transformational analogy as adaptation method, to adapt to problem q the plan of the recovered case c. In the contexts of supervised learning, solutions of both q and c are known and it is possible to find the amount of steps in which both q and c differs. It is common the use of the edit distance (also known as Leveshtein distance [19]) to measure the difference between two sequences. Such distance (denoted here as d_{LEV}) is defined as the minimum cost to transform a sequence into another one, given a set of edit operations and an associated cost to each operation. The set of edit operations used in most of the applications includes the insertion, the deletion and the replacement. Given two planning problems c_i and c_j with known solutions (plans) s_i and s_j, the similarity of their solutions (*useful*) can be expressed as a dual function of the edit distance:

$$\sigma_s(s_i, s_j) = util(s_i, s_j) = 1 - \frac{d_{LEV}(s_i, s_j)}{|s_i| + |s_j|} \tag{7}$$

where $|s_i|$ expresses the amount of steps in the plan solution s_i.

Note that if the plans are identical, then $_S(s_i,s_i) = 1$; on the other hand, if they differ in all the steps, the value is 0.

Therefore, the hypothesis to demonstrate in this experiment is the following:

H1: In the elaboration of the manufacturing plans the use of a weighted similarity function, by applying the proposed algorithm for weights calculation, allows the recovering of plans that require less adaptation effort.

4.2 Experimental Settings

In the following experiment a case base of 130 cases is used, where each case keeps the description of the piece and its manufacturing plan. This information was gathered from the conventional production system, where a process planner studies a new piece, model or plane (drawing); and then determines the appropriate procedures to produce it (plan). That is, for each case of this CB, a solution given by the experts is known a priori.

Considering each case of the CB as a problem q to solve, two different orders from the remaining cases are obtained:

- Considering the value of case similarity with problem q, according to the similarity function that appears in the expression (1):

$$O_{sim}(q) = \{c_1, c_2, \ldots, c_n\}$$

- Considering the utility of the solution of the case for solving problem q, according to (7):

$$O_{util}(q) = \{c_{\pi_1}, c_{\pi_2}, \ldots, c_{\pi_n}\}$$

As far as the orders $O_{sim}(q)$ and $O_{util}(q)$ agree (smaller "disorder"), then "the more similar" cases recovered for q are "more useful" to reach the solution of q, and therefore to reuse them would be more efficient since it requires a lesser effort in the adaptation. The disorder's measurement is the one that appears in expression (2).

In order to prove the hypothesis that was previously specified, in the recovery of the most similar cases to problem q by using the similarity function (*sim*), two variants are depicted:

1. Feature weights are not considered, that is, all of them have the same importance (we will refer to it as non-weighted similarity)
2. The weights obtained with the algorithm previously proposed are considered. (weighed similarity)

For each case q there exist two measures of "associated disorder". Accordingly with the previously defined choices, two continuous variables arise: $D_{sim}{}^{I}$ (variant I) and $D_{sim}{}^{w}$ (variant II).

4.3 Results

The statistical processing refers to a comparison of 2-related samples denotes by variables ($D_{sim}{}^{I}$ and $D_{sim}{}^{w}$), considering two different moments according to the two variants explained above. From the statistical point of view, it is desirable to reject the hypothesis that the two variables have the same distribution (equality of positive and negative ranks in the differences of observed variables).

In order to do this, a non parametric test was used: the Wilcoxon signed-rank test. To improve accuracy in the Wilcoxon test significance, Monte Carlo simulation techniques are used. The significance is based in 10000 sampled tables and a 99% confidence interval for the significance can be estimated.

The results of the comparison are as follows. As it is shown in the Table 1, the disorder measured in the calculation of *sim* with the variant II is less than the disorder observed with variant I in 99 from 130 cases; and there aren't ties. Then, the Wilcoxon test determines significant advantages of the weighted similarity (Significance: 0.000 with confidence interval close to this value (Table 2).

So, concerning the hypothesis *H1* we can conclude that if we use a weighted similarity and the weights are computed according to the proposed algorithm, then those plans that lead to the most similar retrieved cases require less effort for adaptation in new problems solving.

Table 1. Results from Wilcoxon Signed Rank Test

Ranks in Wilcoxon Signed Ranks Test

		N	Mean Rank	Sum of Ranks
Disorder with weigthed similarity vs no- weighted	Negative Ranks	119[a]	69.78	8304.00
	Positive Ranks	11[b]	19.18	211.00
	Ties	0[c]		
	Total	130		

a. Disorder with weighted similarity < Disorder with no-weighted similarity

b. Disorder with weighted similarity > Disorder with no-weighted similarity

c. Disorder with weighted similarity = Disorder with no-weighted similarity

Table 2. Results from Wilcoxon Test (confidence)

Wilcoxon Test Statistics[a]

			Weighted vs no-weighted
Z			-9.043
Monte Carlo Sig. (2-tailed)	Sig.		.000
	99% Confidence Interval	Lower Bound	.000
		Upper Bound	.000

a. Based on 10000 sampled tables with starting seed 2000000.

5 Conclusions

The definition of a similarity metric is one of the most important areas in the development of a cases-based system. As it is shown in the epigraph 2 several studies have proved that the effectiveness of the system strongly depends on the correct definition of the similarity approach. To reduce this dependence, the similarity functions include the importance of the features. Unfortunately, most of the approaches to learn this importance are applied to classification tasks where a method of adaptation is not needed. In other cases, the mechanisms of learning are dependent of the application domain.

In this work, the similarity between solutions is used as a heuristic to estimate the features' importance and it is presented an experiment in the domain of the cases-based planning that shows the effectiveness of such approach.

References

1. Dasarathy B.V, ED. (1990) Nearest neighbor norms: NN Pattern ClassificationTechniques. Los Alamitos, California: IEEE Computer Society Press
2. Mitchell T.M., Machine Learning. McGraw-Hill 1997

3. Aamod, A. y Plaza, E.. Case-based reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, vol.7, no.1, 1994
4. D.Wettschereck and D.W.Aha. Weighting features. In Proceeding of the 1st International Conference on Case-Based Reasoning. Springer, 1995.
5. Hanks, S. and Weld, D.S. A domain-independent algorithm for Plan adaptation. Journal of Artificial Intelligence Research 2, pp. 319-360. 1995
6. Wilke, W. & Bergmann, R. (1997). Techniques and knowledge used for adaptation during case-based problem solving. In 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems.
7. Leake, D.. Case-based reasoning: Experiences, Lessons and Future Direction, AAAI Press/ MIT Press, 1996
8. Leake D. B.. Learning Adaptation Strategies by Introspective Reasoning about Memory Search. In David Leake, editor, Proceedings AAAI-93 Workshop on Case- Based Reasoning, AAAI Press, pp. 57-63. 1993
9. Smyth, B. And Keane, M.T.. Retrieving adaptable cases. In Wess, S. Et al. (Eds.), Topics in Case-based reasoning: First European Workshop, EWCBR-93, selected papers, vol. 837 of Lectures Notes in Artificial Intelligence, pp. 209-220. Springer. 1994
10. Bergmann, R. And Wilke, W.. Towards a new formal model of transformational adaptation in Case-based reasoning. In Proceedings of European Conference on Artificial Intelligence ECAI'98, H. Prade (Ed.). John Wiley and Sons, 1998
11. Smyth, B. Et al.. Chapter 9 In Case-based reasoning: Experiences, Lessons and Future Direction, Leake, D. (Ed.), AAAI Press/ MIT Press, 199
12. Mille, A. et al.. A unifying framework for Adaptation in case-based reasoning. ECAI 96 12th European Conference on Artificial intelligence, W. Wahlster (Ed.). John Wiley & Sons. 1996
13. Smyth, B. And Keane, M.T.. Experiments on adaptation-guided retrieval in case-based design. In Proceedings of the First International Conference on Case-Based Reasoning, Portugal, pp. 313-324. 1995
14. Goel, A.. Integration of Case-Based Reasoning and Model-Based reasoning for adaptive design problem solving. PhD thesis. Ohio State University. 1989
15. Brand, M. et al.. A model-based approach to the construction of adaptative case-based planning systems. In Proceedings of the Case Based Reasoning Workshop, Florida, USA, 1989
16. Fox, S. and Leake, D.. Modeling case-based planning for repairing reasoning failures. In Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms. AAAI Press. Technical report WS-95-05
17. Ihrig, L. and Kambhampati, S. (1995). An explanation-based approach to improve retrieval in case-based planning. Current Trends in AI Planning: EWSP'95 IOS Press
18. Bergmann, R., Richter, M.M., Schmitt, S., Stahl, A., Vollrath, I.: Utility-Oriented Matching: A New Research Direction for Case-Based Reasoning. 9th German Workshop on Case-Based Reasoning (GWCBR'2001), 2001.
19. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, 10:707–710, 1966

Epistemic Actions and Ontic Actions: A Unified Logical Framework

Andreas Herzig and Tiago De Lima*

IRIT, Université Paul Sabatier, Toulouse, France

Abstract. We present a reasoning about actions framework based on a sum of epistemic logic S5 and propositional dynamic logic PDL together with a 'no forgetting' principle, also called perfect recall. We show that in our framework an action may be decomposed into a purely ontic action followed by a purely epistemic action. We also show that the latter is completely definable in terms of simple observations, i.e., 'test that' actions and that they are equivalent to public announcements of public announcement logic PAL as studied by Plaza, van Benthem and others. Finally, since these actions respect 'no learning' principle we show that a unified reduction method based on regression, as studied by Reiter, applies.

1 Introduction

We present a general framework allowing to reason about actions in the case where the agent has no complete information about the state of the world. In this case the agent must be able to perform not only physical, STRIPS-like actions (that we call here ontic actions), but also epistemic actions that allow to acquire information about the state of the world. Such actions include observations (learning that a proposition is true) and tests (sensing actions).

We are interested in mono-agent environments where all action laws are known and events are public. The domain described below, taken from [1], is an example.

Example 1 (Princess). The domain consists of 2 doors. Behind one of the doors there is a tiger and behind the other one there is a princess. The agent does not know where tiger and princess are. The available actions are the following.

- $listen(i)$: listen to what happens behind door i, which results in hearing the tiger roaring if there is one behind the door;
- $open(i)$: open door i, which results in marrying the princess or being eaten by the tiger, depending on what is behind the door.

The agent's goal is to marry the princess and to stay alive. Notice that $listen(i)$ is an epistemic action and $open(i)$ is an ontic action. A possible plan to achieve that is the sequence of actions: $listen(1)$; $\mathsf{if}\, K\, tiger(1) \,\mathsf{then}\, open(2) \,\mathsf{else}\, open(1)$. □

* Supported by the Programme Alßan, the European Union Programme of High Level Scholarships for Latin America, scholarship number E04D041703BR.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 409–418, 2006.

Our aim in this paper is to design the simplest framework able to deal with such examples. This framework will be in terms of an epistemic dynamic logic EDL. It is the sum of S5 logic for modelling knowledge and propositional dynamic logic (without '$*$') for modelling actions, together with a perfect recall ('no forgetting') axiom. The latter says that if the agent knows the effects of an action, then after execution of that action the agent knows the effects.

After introducing syntax and semantics of EDL in Section 2 we show, in Section 3, that in EDL every action can be decomposed into an ontic action followed by an epistemic action. We then show that epistemic actions can be reduced to sequences of observations in Section 4. It turns out that the latter are nothing but announcements of public announcement logic PAL (as studied by Plaza [2], Gerbrandy [3], Baltag [4], van Benthem [5] and van Ditmarsch [6]). In Section 5 we briefly recall the dynamic logic regression method. In Section 6 we show that under some reasonable hypothesis the reduction axiom based proof method for PAL can be combined with Reiter's regression into a proof method for our general framework. Finally, in Section 7 we present some discussions and the conclusion of the paper.

2 Preliminaries

Definition 1. *Let P be a countably infinite set of propositional letters. The* language of epistemic dynamic logic, *$\mathcal{L}_{\mathrm{EDL}}$, is defined by the following BNF.*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi \mid [\alpha]\varphi$$

where p ranges over P and α belongs to the action language defined as follows. Let A be a countably infinite set of (abstract) actions. The action language $\mathcal{L}(A)$ *is defined by the following BNF.*

$$\alpha ::= \mathsf{skip} \mid a \mid \mathsf{if}\ \varphi\ \mathsf{then}\ \alpha\ \mathsf{else}\ \alpha \mid \alpha;\alpha \mid \alpha \cup \alpha$$

where a ranges over A.

The formula $K\varphi$ is read 'agent knows that φ' and the formula $[\alpha]\varphi$ is read 'φ holds after every execution of α'. The construction $\alpha_1;\alpha_2$ is the sequential composition of α_1 and α_2, and the construction $\alpha_1 \cup \alpha_2$ is the nondeterministic choice between α_1 and α_2. To simplify notation, we use the common abbreviations for $\vee$, $\rightarrow$, $\leftrightarrow$, $\langle\alpha\rangle$ (we recall that $\langle\alpha\rangle\varphi = \neg[\alpha]\neg\varphi$). In the sequel, two fragments of $\mathcal{L}_{\mathrm{EDL}}$ are important. The first one contains no modal operators. These formulae are called *boolean formulae* and we note this fragment $\mathcal{L}_{\mathrm{PL}}$. The second one contains $\mathcal{L}_{\mathrm{PL}}$ and also formulae with the operator K. These formulae are called *epistemic formulae* and we note this fragment $\mathcal{L}_{\mathrm{S5}}$.

This language permits description of effect and executability laws. For example, the formula $\mathit{tiger}(1) \rightarrow [\mathit{open}(1)]\neg\mathit{alive}$ means that if there is a tiger behind door 1, then the agent will die after opening it. The formula $\langle\mathit{open}(1)\rangle\top \leftrightarrow \mathit{alive}$ means that action $\mathit{open}(1)$ is executable if and only if the agent is alive. Also conditional plans are possible such as $\mathit{listen}(1); \mathsf{if}\ K\,\mathit{tiger}(1)\ \mathsf{then}\ \mathit{open}(1)\ \mathsf{else}\ \mathit{open}(2)$.

Definition 2. *An* EDL-model *is a tuple of the form* $M = \langle W, R, T, V \rangle$ *where*

- W *is a non-empty set of possible worlds;*
- $R \subseteq \wp(W \times W)$ *is a reflexive, symmetric and transitive relation.*
- $T : \mathcal{L}(A) \to \wp(W \times W)$ *is a set of transitions, noted* T_α*.*
- $V : P \to \wp(W)$ *associates an interpretation to each* $p \in P$*, noted* $V(p)$*.*
- *for all* $\alpha \in \mathcal{L}(A) : T_\alpha \circ R \subseteq R \circ T_\alpha$*.*

For convenience, we define $R(w) = \{w' : (w, w') \in R\}$ *and* $T_\alpha(w) = \{w' : (w, w') \in T_\alpha\}$*. The relation* R *models the epistemic state of the agent:* $R(w)$ *is the set of possible worlds the agent considers to be possible at* w*. The transitions* T_α *model actions:* $T_\alpha(w)$ *is the set of worlds resulting from the execution of* α *in* w*.*

The interaction constraint makes that the worlds the agent considers possible after execution of α in w, i.e. $(T_\alpha \circ R)(w)$, are a subset of the outcomes of α that the agent considers possible at w, i.e. $(R \circ T_\alpha)(w)$. Such a constraint is called *perfect recall* in [7] and *no forgetting* in [1].

Definition 3. *The satisfaction relation* $\Vdash$ *is defined as follows.*

$$\begin{array}{ll}
M, w \Vdash p & \text{iff } w \in V(p) \\
M, w \Vdash \neg\varphi & \text{iff } M, w \nVdash \varphi \\
M, w \Vdash \varphi_1 \wedge \varphi_2 & \text{iff } M, w \Vdash \varphi_1 \text{ and } M, w \Vdash \varphi_2 \\
M, w \Vdash K\varphi & \text{iff for all } w' \in R(w),\ M, w' \Vdash \varphi \\
M, w \Vdash [\alpha]\varphi & \text{iff for all } w' \in T_\alpha(w),\ M, w' \Vdash \varphi
\end{array}$$

Definition 4. *An* EDL-standard model *is a model such that*

$$\begin{array}{ll}
T_{\mathsf{skip}} & = \{(w, w) : w \in W\} \\
T_{\mathsf{if}\ \varphi\ \mathsf{then}\ \alpha_1\ \mathsf{else}\ \alpha_2} & = \{(w, w') : M, w \Vdash \varphi \text{ implies } (w, w') \in T_{\alpha_1} \text{ and} \\
 & \qquad M, w \nVdash \varphi \text{ implies } (w, w') \in T_{\alpha_2}\} \\
T_{\alpha_1;\alpha_2} & = T_{\alpha_1} \circ T_{\alpha_2} \\
T_{\alpha_1 \cup \alpha_2} & = T_{\alpha_1} \cup T_{\alpha_2}
\end{array}$$

Definition 5. *A formula* φ *is* true *in* M*, noted* $M \Vdash \varphi$*, if and only if* $M, w \Vdash \varphi$ *for all* $w \in W$*. A formula* φ *is* valid*, noted* $\models \varphi$*, if and only if* $M \Vdash \varphi$ *for all standard models* M*. A formula* φ *is a* valid consequence *of a set of formulae* Σ*, noted* $\Sigma \models \varphi$*, if and only if* $M \Vdash \psi$ *for all* $\psi \in \Sigma$ *implies* $M \Vdash \varphi$ *for all standard models* M*.*

No specific conditions for T_a are given. It means that their behaviour must be defined with formulae in $\mathcal{L}_{\mathrm{EDL}}$. For instance, for the Example 1 we must write the set of formulae Δ that allows to infer action laws such as

$$\begin{array}{rr}
tiger(i) \to [listen(i)]K\, tiger(i) & princess(i) \to [open(i)]married \\
\neg tiger(i) \to [listen(i)]K\neg tiger(i) & tiger(i) \to [open(i)]\neg alive \\
\langle listen(i)\rangle\top \leftrightarrow alive & \langle open(i)\rangle\top \leftrightarrow alive\ .
\end{array}$$

A solution to this planning problem is an action (or plan) π such that

$$\Delta \models (alive \wedge \neg married) \rightarrow \langle \pi \rangle (alive \wedge married) \ .$$

One can prove that it holds for $\pi = listen(1); \mathsf{if}\, K\, tiger(1) \,\mathsf{then}\, open(2) \,\mathsf{else}\, open(1)$. It follows from standard results in modal logic on Sahlqvist formulae [8] that EDL has a sound and complete proof system made up of the standard Modus Ponens and Necessitation inference rules for K and every $[\alpha]$, plus the following axiom schemes.

K(K).	$K(\varphi_1 \rightarrow \varphi_2) \rightarrow (K\varphi_1 \rightarrow K\varphi_2)$
T(K).	$K\varphi \rightarrow \varphi$
4(K).	$K\varphi \rightarrow KK\varphi$
5(K).	$\neg K\varphi \rightarrow K\neg K\varphi$
K(α).	$[\alpha](\varphi_1 \rightarrow \varphi_2) \rightarrow ([\alpha]\varphi_1 \rightarrow [\alpha]\varphi_2)$
Def(skip).	$[\mathsf{skip}]\varphi \leftrightarrow \varphi$
Def(if).	$[\mathsf{if}\ \varphi_1\ \mathsf{then}\ \alpha_1\ \mathsf{else}\ \alpha_2]\varphi_2 \leftrightarrow ((\varphi_1 \rightarrow [\alpha_1]\varphi_2) \wedge (\neg\varphi_1 \rightarrow [\alpha_2]\varphi_2))$
Def(;).	$[\alpha_1;\alpha_2]\varphi \leftrightarrow [\alpha_1][\alpha_2]\varphi$
Def($\cup$).	$[\alpha_1 \cup \alpha_2]\varphi \leftrightarrow [\alpha_1]\varphi \wedge [\alpha_2]\varphi$
NF.	$K[\alpha]\varphi \rightarrow [\alpha]K\varphi$

The first four are the standard axioms of S5 logic, and the following four are standard axioms of PDL. NF is the 'no forgetting' axiom: if the agent knows that φ holds after α, then he indeed knows φ after α.

EDL is decidable and expressive enough for our purpose. However, for practical applications, it is not as suitable as desired. The complexity of the validity problem in EDL is PSPACE-hard [1].

In this paper we go beyond EDL. First, we prove that every action can be decomposed in a purely ontic followed by a purely epistemic action. Next, we propose a refined analysis of the concept of epistemic actions. This results in a stronger logic, which in particular augments EDL by a *no learning* principle.

3 The Decomposition Theorem

Roughly, purely ontic actions stand for actions that do not involve any perception: the agent only knows that action α has been performed, without learning about its (possibly nondeterministic or conditional) effects. On the other hand, purely epistemic actions cannot change facts about the world. Typical examples are sensing actions (testing whether a proposition is true or not) and observations (learning that a proposition is true). The definitions below make precise this distinction.

Definition 6. *In a model M, $o \in A$ is* purely ontic *if and only if T_o satisfies the following properties.*

- Epistemic determinism*: if $w_1, w_2 \in T_o(w)$, then $R(w_1) = R(w_2)$.*
- No learning*: if $w' \in (R \circ T_o)(w)$ and $T_o(w) \neq \emptyset$, then $w' \in (T_o \circ R)(w)$.*

The epistemic determinism constraint corresponds to the fact that the agent cannot distinguish between nondeterministic outcomes of an action: whether the coin falls heads or tails, the agent only knows that a coin has been tossed and that the disjunction holds. The 'no learning' constraint corresponds to the fact that if the agent considers that w' is a possible outcome of execution of o in w, then the agent keeps on considering w' to be a possible world after o.

Definition 7. *In a model M, $e \in A$ is* purely epistemic *if and only if it satisfies the following property.*

– Preservation: *If $w' \in T_e(w)$, then for all $p \in P$, ($w \in V(p)$ iff $w' \in V(p)$).*

This constraint corresponds to the fact that epistemic actions do not change the facts about the world.

Now, we show that these two kinds of actions is all we need: every transition relation can be decomposed appropriately.

Theorem 1 (Decomposition). *Let $\alpha \in \mathcal{L}(A)$ and let $\varphi \in \mathcal{L}_{\mathrm{EDL}}$. The formula φ is satisfiable if and only if there exist actions o and e such that: o is purely ontic, e is purely epistemic, and $\varphi[(e;o)/\alpha]$ (the formula obtained by replacing $e;o$ for α in φ) is satisfiable.*

Proof (sketch). From right to left is straightforward. From left to right is established by introducing intermediate worlds that correspond to the outcome of action o. □

It enables us to make a partition in the set A of abstract actions. From now on, it is formed by the union of two disjoint sets: A^e of purely epistemic and A^o of purely ontic actions.

4 How Many Kinds of Epistemic Actions Are There?

We start our analysis by considering the most basic kind of epistemic action: *observations*[1]. It can roughly be understood as an exogenous event that makes the agent observe that φ. The observation that φ holds is noted $\mathsf{obs}\,\varphi$. Then, the formula $[\mathsf{obs}\,\varphi]\psi$ is read 'ψ holds after the agent observes that φ'. We now consider a variant of EDL where the only epistemic actions are observations. We therefore restrict our attention to the following language.

Definition 8. *Let o range over A^o. The language of* $\mathrm{EDL}^{\mathsf{obs}}$ *$\mathcal{L}_{\mathrm{EDL}^{\mathsf{obs}}}$, is the same as given in Definition 1, with the difference that $A^e = \emptyset$ and actions α are now elements of the* ontic action language *$\mathcal{L}(A^o)$, defined according to the following BNF.*

$$\alpha ::= \mathsf{skip} \mid o \mid \mathsf{obs}\,\varphi \mid \mathsf{if}\ \varphi\ \mathsf{then}\ \alpha\ \mathsf{else}\ \alpha \mid \alpha;\alpha \mid \alpha \cup \alpha$$

[1] This action is also named *test that* in some different approaches.

Definition 9. *An* EDL$^{\mathsf{obs}}$-standard model *is an* EDL*-standard model such that every* $T_{\mathsf{obs}\,\varphi}$ *satisfies the following constraints:*

- *If* $w' \in T_{\mathsf{obs}\,\varphi}(w)$ *then for all* $p \in P$, $(w \in V(p)$ *iff* $w' \in V(p))$;
- *If* $M, w \nVdash \varphi$ *then* $T_{\mathsf{obs}\,\varphi}(w) = \emptyset$;
- *If* $M, w \Vdash \varphi$ *then exists* $w' \in T_{\mathsf{obs}\,\varphi}(w)$ *and for all* $w'' \in T_{\mathsf{obs}\,\varphi}(w)$, w' *and* w'' *satisfy the same formulae;*
- *If* $w' \in T_{\mathsf{obs}\,\varphi}(w)$ *then* $R(w') = (R \circ T_{\mathsf{obs}\,\varphi})(w)$.

The first constraint says that observations are purely epistemic actions. The second constraint says that truth of φ is a necessary condition for executability of $\mathsf{obs}\,\varphi$. The third condition says that observations are deterministic. And the fourth condition says that observations are also ontic actions.

It can be shown that the following validities characterise observations (note that the last one corresponds to the 'no learning' principle).

$$\begin{array}{ll}
\mathrm{Pre}(\mathsf{obs}). & \Phi \rightarrow [\mathsf{obs}\,\varphi]\Phi \quad \text{for a boolen formula } \Phi \\
\mathrm{Exe}(\mathsf{obs}). & \varphi \leftrightarrow \langle \mathsf{obs}\,\varphi\rangle\top \\
\mathrm{Det}(\mathsf{obs}). & \langle \mathsf{obs}\,\varphi\rangle\psi \rightarrow [\mathsf{obs}\,\varphi]\psi \\
\mathrm{NL}(\mathsf{obs}). & [\mathsf{obs}\,\varphi]K\psi \rightarrow ([\mathsf{obs}\,\varphi]\bot \vee K[\mathsf{obs}\,\varphi]\psi)
\end{array}$$

However, other kinds of epistemic actions exist. For instance, *listen*(1) (Example 1) is not an observation. It is what we call a *test*[2]: given a formula φ it returns whether φ holds or not. We note this kind of action $\mathsf{test}\,\varphi$, and the formula $[\mathsf{test}\,\varphi]\chi$ is read 'χ holds after the agent tests whether φ'. For example, *listen*(1) can be written '$\mathsf{test}\ \mathit{tiger}(1)$'.

In fact, *listen*(1) is conditional: the test depends on the context. It is noted, $\mathsf{test}\,\varphi\ \mathsf{if}\ \psi$, where ψ is the condition for the test whether φ. The same may also be applied to observations. Then, the formula $[\mathsf{obs}\,\varphi\ \mathsf{if}\ \psi]\chi$ is read 'under the condition ψ, χ holds after the agent observes that φ'. For example, *listen*(1) is $\mathsf{test}\ \mathit{tiger}(1)\ \mathsf{if}\ \mathit{alive}$. We can see these constructions as abbreviations:

$$\begin{array}{lll}
\langle \mathsf{obs}\,\varphi\ \mathsf{if}\ \psi\rangle\chi & \stackrel{def}{=} & \psi \wedge \langle \mathsf{obs}\,\varphi\rangle\chi \\
\langle \mathsf{test}\,\varphi\rangle\chi & \stackrel{def}{=} & \langle \mathsf{obs}\,\varphi\rangle\chi \vee \langle \mathsf{obs}\,\neg\varphi\rangle\chi \\
\langle \mathsf{test}\,\varphi\ \mathsf{if}\ \psi\rangle\chi & \stackrel{def}{=} & \psi \wedge \langle \mathsf{test}\,\varphi\rangle\chi
\end{array}$$

We leave to the reader the confirmation that all these definitions match the intuitions behind the actions introduced above.

We want to have all kinds of epistemic actions in our logic. Tests and conditional observations are definable in terms of unconditional observations. Still better, in the theorem below, we prove that every purely epistemic action can be defined in terms of this operator.

Theorem 2 (Observations are general). *Suppose that* P *and* A *are finite. Let* $\varphi \in \mathcal{L}_{\mathrm{EDL}}$ *and let* e *be a purely epistemic action. The formula* φ *is satisfiable in finite models if and only if there exists a (complex) observation* ϵ *such that* $\varphi[\epsilon/e]$ *is satisfiable in finite models.*

[2] This action is also named *test if* or *sense* in the literature.

Proof (sketch). From right to left: since ϵ is purely epistemic, take $T_e = T_\epsilon$. From left to right: suppose $M, w \Vdash \varphi$. It is shown in [5] that every $w \in W$ can be characterised by a formula $\delta(w)$ such that $M, w \Vdash \delta(w')$ iff w and w' satisfy the same formulae. Now, it can be shown that $T_e = T_{\bigcup_{w \in W}(\mathsf{obs}\,\delta(w)\,\mathsf{if}\,\gamma(w))}$, where $\gamma(w) = \bigvee_{v \in (T_e \circ R \circ T_e^{-1})(w)} \delta(v)$ by using the fact that: $\forall w, v, v' \in W$, if $v \in T_{\mathsf{obs}\,\gamma(w)}(w)$ and $v' \in T_e(w)$ then v and v' satisfy the same epistemic formulae. For a detailed proof the reader can refer to [9]. □

In other words, in EDL, sequences and nondeterministic compositions of observations suffice to express every kind of purely epistemic action.

In fact, the operator obs turns out to be the same as '!' of the public announcement logic PAL, originally proposed in [2]. Both PAL and EDL$^{\mathsf{obs}}$ have the same set of axioms for the operators '!' and 'obs' respectively. Here, we use a mono-agent version of this logic. Some recent works [6, 10] show that PAL is very suitable for modelling multi-agent communication. In addition, it is possible to incorporate other notions such as belief and common knowledge. PAL is itself a special case of other communication logics proposed, for instance, in [4, 3, 10]. This suggests that EDL$^{\mathsf{obs}}$ can be extended to handle all these notions.

5 Describing Deterministic Actions

As shown in [11], purely ontic actions that are deterministic can be eliminated under the following two hypotheses: Δ does not contain static laws and the agent has complete information about executability conditions and action effects. A method for performing this elimination in dynamic logic was proposed in [12].

From now on, we make the assumption that all actions in A^o are deterministic: for every model M and for every $o \in A^o$, $T_o(w)$ is a function. Then, we add the following axiom scheme to EDL$^{\mathsf{obs}}$. Let $o \in A^o$.

$$\mathrm{Det(o)}. \quad \langle o \rangle \varphi \rightarrow [o]\varphi$$

The hypothesis of complete information enables descriptions of actions in the following format.

Definition 10. *Let P be the finite set of propositional letters of the domain, let A^o be the finite set of abstract ontic actions of the domain, and let o range over A^o. An* action description *of o is a tuple $D(o)$ of the form $\langle Poss, Effect^+, Effect^-, Cond^+, Cond^- \rangle$ such that*

- *$Poss \in \mathcal{L}_{\mathrm{S5}}$ is the executability precondition, noted $Poss(o)$;*
- *$Effect^+ \subseteq P$ is the set of all possible positive effects, noted $Effect^+(o)$;*
- *$Effect^- \subseteq P$ is the set of all possible negative effects, noted $Effect^-(o)$;*
- *$Cond^+ : Effect^+ \rightarrow \mathcal{L}_{\mathrm{PL}}$ assigns to each $p \in Effect^+(o)$ a boolean formula describing its positive precondition, noted $Cond^+(o, p)$; and*
- *$Cond^- : Effect^- \rightarrow \mathcal{L}_{\mathrm{PL}}$ assigns to each $p \in Effect^-(o)$ a boolean formula describing its negative precondition, noted $Cond^-(o, p)$.*

In addition, $Cond^+$ and $Cond^-$ must satisfy: $\models \neg(Cond^+(o,p) \wedge Cond^-(o,p))$.

In the sequel, we also assume that this kind of description is given for observations. In this case, we set $Poss(\mathsf{obs}\varphi) = \varphi$, and $Effect^+(\mathsf{obs}\varphi) = Effect^-(\mathsf{obs}\varphi) = Cond^+ = Cond^- = \emptyset$.

Now, suppose that an action description of an action a is given. Then the set of laws of a, $\Delta(a)$, can be automatically generated as follows:

1. Add the following executability axiom to $\Delta(a)$: $Poss(a) \leftrightarrow \langle a\rangle\top$.
2. For every $p \in Effect^+(a)$, add the following two positive effect axioms to $\Delta(a)$: $Cond^+(a,p) \rightarrow [a]p$ and $(\neg Cond^+(a,p) \wedge \neg p) \rightarrow [a]\neg p$.
3. For every $p \in Effect^-(a)$, add the following two negative effect axioms to $\Delta(a)$: $Cond^-(a,p) \rightarrow [a]\neg p$ and $(\neg Cond^-(a,p) \wedge p) \rightarrow [a]p$.
4. For every $p \notin Effect^+(a)$, add the following frame axiom to $\Delta(a)$: $\neg p \rightarrow [a]\neg p$.
5. For every $p \notin Effect^-(a)$, add the following frame axiom to $\Delta(a)$: $p \rightarrow [a]p$.

6 Regression in EDL$^{\mathsf{obs}}$

Planning under incomplete knowledge through regression was originally proposed in [11, 13] using situation calculus as the base formalism. Situation calculus is a dialect of second order logic, but through regression, a semi-decidable procedure for planning is possible. The corresponding mechanism for our logic should act as follows: given a problem of the form $\Delta \models \iota \rightarrow \langle\alpha\rangle\gamma$, then, under some reasonable assumptions, return an equivalent problem of the form $\models \iota \rightarrow \mathrm{reg}(\langle\alpha\rangle\gamma)$ where $\mathrm{reg}(\langle\alpha\rangle\gamma)$ is a "simplified" version of the original formula, namely, without dynamic operators. Hence, the result is a formula in S5 logic. Validity in S5 is well known to be NP-Complete. Therefore, although a decision procedure for EDL is already known, regression based planning is still interesting because in practice it is often more efficient.

Definition 11. *Let $\varphi, \varphi_1, \varphi_2 \in \mathcal{L}_{\mathrm{EDL}^{\mathsf{obs}}}$, let $\Phi \in \mathcal{L}_{\mathrm{S5}}$, let $\alpha_1, \alpha_2 \in \mathcal{L}(A^o)$, let $b \in (A^o \cup \bigcup_{\varphi\in\mathcal{L}_{\mathrm{S5}}} \mathsf{obs}\,\varphi)$ and let $p \in P$. The* EDL regression operator reg() *is inductively defined as follows.*

1. $\mathrm{reg}(\Phi) = \Phi$;
2. *if* $p \notin Effect^+(b)$ *and* $p \notin Effect^-(b)$, *then* $\mathrm{reg}([b]p) = \neg Poss(b) \vee p$;
3. *if* $p \notin Effect^+(b)$ *and* $p \in Effect^-(b)$, *then* $\mathrm{reg}([b]p) = \neg Poss(b) \vee (p \wedge \neg Cond^-(b,p))$;
4. *if* $p \in Effect^+(b)$ *and* $p \notin Effect^-(b)$, *then* $\mathrm{reg}([b]p) = \neg Poss(b) \vee Cond^+(b,p) \vee p$;
5. *if* $p \in Effect^+(b)$ *and* $p \in Effect^-(b)$, *then* $\mathrm{reg}([b]p) = \neg Poss(b) \vee Cond^+(b,p) \vee (p \wedge \neg Cond^-(b,p))$;
6. $\mathrm{reg}([b]\neg\varphi) = \neg Poss(b) \vee \neg\,\mathrm{reg}([b]\varphi)$;
7. $\mathrm{reg}([b](\varphi_1 \wedge \varphi_2)) = \mathrm{reg}([b]\varphi_1) \wedge \mathrm{reg}([b]\varphi_2)$;
8. $\mathrm{reg}([b]K\varphi) = \neg Poss(b) \vee K\,\mathrm{reg}([b]\varphi)$;

9. $\mathrm{reg}([\mathsf{skip}]\varphi) = \mathrm{reg}(\varphi)$;
10. $\mathrm{reg}([\mathsf{if}\ \varphi\ \mathsf{then}\ \alpha_1\ \mathsf{else}\ \alpha_2]\psi) = \mathrm{reg}((\varphi \to [\alpha_1]\psi) \wedge (\neg\varphi \to [\alpha_2]\psi))$;
11. $\mathrm{reg}([\alpha_1;\alpha_2]\varphi) = \mathrm{reg}([\alpha_1]\,\mathrm{reg}([\alpha_2]\varphi))$; *and*
12. $\mathrm{reg}([\alpha_1 \cup \alpha_2]\varphi) = \mathrm{reg}([\alpha_1]\varphi) \wedge \mathrm{reg}([\alpha_2]\varphi)$;

By using an appropriate reduction ordering one can prove that the rewriting procedure reg() is well-defined and terminates.

Theorem 3 (EDL regression). *Let* $B = A^o \cup \bigcup_{\varphi\in\mathcal{L}_{\mathrm{S5}}} \mathsf{obs}\,\varphi$. *Let* $\Delta(B) = \bigcup_{b\in B} \Delta(b)$ *be the action theory corresponding to action descriptions in terms of Poss, Effect*$^+$*, Effect*$^-$*, Cond*$^+$ *and Cond*$^-$. *If* φ *is an* EDL$^{\mathsf{obs}}$ *formula, then* $\Delta(B) \models \varphi$ *if and only if* $\models_{\mathrm{S5}} \mathrm{reg}(\varphi)$.

Proof (sketch). Equalities from 1 to 5 for ontic actions are proved to be logical equivalences under $\Delta(B)$ in [12]. The proof for observations is the same. Equalities from 9 to 12 are proved by using the axiom schemes Def(skip), Def(if), Def(;) and Def(∪). For ontic actions, equalities from 6 to 8 are proved by using Definition 6 and for observations they are proved by the following validities that can be deduced from our axioms for obs . Notice that the last one corresponds to both 'no forgetting' and 'no learning' principles for observations.

$[\mathsf{obs}\,\varphi]p \leftrightarrow (\varphi \to p)$
$[\mathsf{obs}\,\varphi]\neg\psi \leftrightarrow (\varphi \to \neg[\mathsf{obs}\,\varphi]\psi)$
$[\mathsf{obs}\,\varphi](\varphi \wedge \chi) \leftrightarrow ([\mathsf{obs}\,\varphi]\psi \wedge [\mathsf{obs}\,\varphi]\chi)$
$[\mathsf{obs}\,\varphi]K\psi \leftrightarrow (\varphi \to K[\mathsf{obs}\,\varphi]\psi)$ □

Corollary 1. $\Delta(B) \models \iota \to [\pi]\gamma$ *if and only if* $\models \iota \to \mathrm{reg}([\pi]\gamma)$.

7 Discussion and Conclusion

Starting with early work by Moore [14], there has been a lot of interest in the reasoning about actions community on the integration of knowledge and sensing actions into the situation calculus since Scherl and Levesque's [13]. Some of these approaches use, without proof, the assumption that actions may be decomposed as shown in the decomposition theorem presented here. In addition, in such approaches, epistemic actions ϵ are characterised by a set $sensedFluents(\epsilon)$. Suppose $sensedFluents(\epsilon) = \{p_1, \ldots, p_n\}$. Then ϵ corresponds to our complex action $\mathsf{obs}\,(p_1 \wedge \ldots \wedge p_n) \cup \mathsf{obs}\,(p_1 \wedge \ldots \wedge \neg p_n) \cup \mathsf{obs}\,(\neg p_1 \wedge \ldots \wedge \neg p_n)$ of nondeterministically observing a boolean combination of $p_1, \ldots, p_n$. Thus such complex actions can be reduced to our simpler, basic observations.

Going beyond these approaches, we allow here for more complex observations of non-boolean, epistemic propositions, such as $\mathsf{obs}\,(p \wedge \neg Kp)$. Such observations are particularly interesting in the multi-agent case that, as said before, is a straightforward extension of EDL$^{\mathsf{obs}}$. We note that contrarily to boolean observations, epistemic observations are not necessarily successful: $[\mathsf{obs}\,(p \wedge \neg Kp)](p \wedge \neg Kp)$ is not valid (and is even unsatisfiable).

Up to now, research in reasoning about actions was quite disconnected from work on public announcement logic PAL that had been done in the 'Dutch tradition' by van Benthem, Gerbrandy, Baltag, van Ditmarsch, Kooi, and others. Just as we do, PAL combines epistemic logic with PDL. Contrarily to us (and contrarily to the AI tradition and terminology), work in PAL has focussed almost exclusively on observations (called public announcements in PAL). From this perspective, we have here extended PAL by ontic actions, connecting it with Reiter's solution to the frame problem. The only similar work is [10], who have augmented PAL by public assignments (of propositional variables to truth values). We are currently investigating the precise relationship.

Acknowledgements. We thank the reviewers for their helpful comments.

References

1. Herzig, A., Lang, J., Longin, D., Polacsek, T.: A logic for planning under partial observability. In: Proc. of the Seventeenth AAAI and the Twelfth IAAI, Austin, The AAAI Press (2000)
2. Plaza, J.: Logics of public communications. In Emrich, M.L., S, P.M., Hadzikadic, M., Ras, Z.W., eds.: Proc. of the Fourth ISMIS, Charlotte (1989) 201–216
3. Gerbrandy, J.: Bisimulations on Planet Kripke. PhD thesis, University of Amsterdam, Amsterdam (1999)
4. Baltag, A., Moss, L.: Logics for epistemic programs. Synthese **139**(2) (2004) 165–224
5. van Benthem, J.: One is a lonely number. In Koepke, P., et al., eds.: Proc. of the Logic Colloquium & Colloquium Logicum, Muenster (2002)
6. van Ditmarsch, H., Kooi, B.: The secret of my success. Synthese (to appear) also available at http://www.cs.otago.ac.nz/research/publications/oucs-2004-10.pdf.
7. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. The MIT Press, Cambridge (1995)
8. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
9. De Lima, T.: Espistemic actions and ontic actions: a unified logical framework (extended version). Technical report, IRIT, Toulouse (2006)
10. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic epistemic logic with assignment. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M., Wooldridge, M., eds.: Proc. of the Fourth AAMAS, Utrecht, ACM (2005) 141–148
11. Reiter, R.: The frame problem in the situation calculus: A simple solution (sometimes) and a completness result for goal regression. In Lifschitz, V., ed.: Papers in Honor of John McCarthy. Academic Press Professional Inc., San Diego (1991) 359–380
12. Demolombe, R., Herzig, A., Varzinczak, I.: Regression in modal logic. Journal of Applied Non-Classical Logics **13**(2) (2003) 165–185
13. Scherl, R., Levesque, H.: The frame problem and knowledge-producing actions. In: Proc. of the Eleventh AAAI, Washington DC, The AAAI Press (1993) 689–695
14. Moore, R.: Reasoning about knowledge and action. Technical Note 191, SRI International (1980)

Strings and Holes: An Exercise on Spatial Reasoning

Pedro Cabalar[1,⋆] and Paulo Santos[2]

[1] Department of Computer Science
Corunna University, Galicia, Spain
cabalar@udc.es
[2] Department of Electrical Engineering
Centro Universitário da FEI, São Paulo, Brazil
psantos@fei.edu.br

Abstract. This paper investigates the challenging problem of encoding the knowledge and reasoning processes involved in the common sense manipulation of physical objects. In particular we provide a formalisation of a domain involving rigid objects, holes and a string within a reasoning about actions and change framework. Therefore, this work investigates the formalisation and reasoning about flexible objects and void space (holes) in a single domain. Preliminary results of automated reasoning within this domain are also presented.

1 Introduction

The field of qualitative spatial reasoning (QSR) [10] attempts the formalisation of spatial knowledge based on primitive relations defined over elementary spatial entities. One of the best known QSR theories, for instance, is the Region Connection Calculus [6], which is a first order axiomatisation of space based on regions and the connectivity relation. Other representations of spatial knowledge include theories about shape [4], distance [3], position [1] amongst others as surveyed in [2]. However, the use of qualitative spatial knowledge within a planning system remains largely neglected.

One possible reason for the lack of problem solving methods handling qualitative spatial knowledge may be connected to the fact that research on QSR has being conducted independently from research on reasoning about actions and change (RAC) and AI planning (apart from exceptions such as [9] and [8]). One of the motivations for the present work is to approximate RAC to reasoning about spatial knowledge by investigating the formalisation and automatic solution of a challenging spatial puzzle.

This paper assumes the puzzle called *The Fisherman's Folly* (Figure 1) that involves spatial entities such as strings, posts, rings, spheres and holes (through the last ones some (but not all) domain objects can pass). The Fisherman's Folly puzzle consists in going from the configuration shown in Figure 1(a) to the configuration in Figure 1(b) by moving the objects positions respecting some domain restrictions. In this sense, the assumed puzzle is similar to the classic 8-puzzle; however, in the present work the domain objects have non-trivial spatial characteristics (such as flexibility and permeability through holes).

The elements of the puzzle are a holed post fixed to a wooden base, a string, a ring, a pair of spheres and a pair of disks. The disks and spheres are attached to the string, along which the latter can move whereby the former is fixed to the string endpoints.

⋆ Partially supported by the Spanish Government CICyT project TIC-2003-9001-C02.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 419–429, 2006.

(a) The initial state. (b) The goal state.

Fig. 1. A spatial puzzle: the Fisherman's Folly

In the initial state (shown in Figure 1(a)) the post is in the middle of the ring, which is supported on the post's base. On the other hand, the string passes through the post's hole in a way that one sphere and one disk remain on each side of the post. It is worth pointing out that the spheres are larger than the post's hole, therefore the string cannot be separated from the post without cutting either the post, or the string, or destroying one of the spheres. The disks and the ring, in contrast, can pass through the post's hole. The goal of this puzzle (depicted in Figure 1(b)) is to find a sequence of transformations of the spatial configuration of the puzzle's objects such that the ring is freed from the system *post-base-string*, maintaining the physical integrity of the domain objects. In fact, the goal state is not fixed to the one shown in Figure 1(b). In order to be considered a solution, it is sufficient to move the ring completely out of the rest of the system, regardless the final configuration of the remaining domain objects.

The complexity imposed by the distinct states of the string allied to the existence of holes in the domain objects makes the formalisation and reasoning about this domain a challenging problem. In order to provide a formal account of the spatial relations involved in the Fisherman's Folly we need to consider in our formalisation (and reasoning processes) the holes in objects, such as the post's hole and the space limited by the ring. This calls for assuming holes as real objects, therefore having the same ontological status as spheres and disks. Reasoning about holes and holed objects has been discussed in detail in [11] from a topological standpoint. However, to the best of our knowledge, the present paper presents the first approach that investigates the problem of how these entities could be engaged in actions.

The string brings a further source of complexity which comes from the related infinity of distinct configurations due to its flexibility. The problem of incorporating knowledge about strings and string manipulation has been tackled in [5] where a robotic system capable of learning to tie a knot from visual observation is proposed, this system is called *the Knot Planning from Observation* (KPO) paradigm. In KPO each state of a string is represented by a matrix encoding the string segments, which are defined by the portion of the string that lies in between its endpoints and points where it crosses over itself. Actions on flexible objects in this context were defined as an extension of the Reidemeister moves in knot theory [7]. This representation is suitable for the identification of string states from a computer vision system; however, it falls short in the context of problem solving, which is the main purpose of the present paper. In this work, we propose a representation for string states that takes into account other objects (including holes) that may be related to the string in the domain. In contrast to the work proposed in [5], we do not take into account knots. Incorporating some of the ideas of the KPO paradigm in our work shall be investigated in the future.

In summary, the purpose of this paper is to investigate the formalisation and autonomous solution of a spatial domain involving holes and a flexible string, contributing with a novel benchmark problem for common sense knowledge representation.

This paper is organised as follows: the next section introduces the formal representation we use for the domain objects; Section 3 discusses the basic actions that operate on the puzzle; some details on a simple Prolog implementation are presented in Section 4; and, finally, Section 5 concludes this paper.

2 Domain Objects

A straightforward classification of the objects in the puzzle would lead to six sorts: the spheres, the disks, the string, the ring, the holed post and the post base. Although the post and its base form a same object, they are clearly distinguishable. In fact, something similar happens with the disks and the string: they form a same "tandem," but a disk and a string clearly have different properties. For the spheres there is no doubt: although limited by the string and the disks in the domain, they are topologically independent objects.

Although the commonsense knowledge about all these sorts can be very rich (for instance, we know that a sphere can roll, a string can form knots, etc) we must focus however on their *relevant* properties for obtaining a satisfactory solution to the puzzle, applying somehow an Occam's razor criterion. Besides, it is also important to fix the very same idea of *satisfactory* solution. Our criterion in this sense is trying to obtain a qualitative description of the movements to be performed, in similar terms to those one could find in a textual description of the solution written in natural language. A related observation is that the same puzzle problem could actually be built with objects of different nature to those commented before. For instance, the spheres and the disks could also be boxes of different sizes, the post base could be a large disk, etc. These small changes are not really essential. However, when a human describes the puzzle solution, she immediately talks about passing objects through *holes*. The puzzle, in fact, deals with four holes: the post hole (ph), the ring hole (r), and the two sphere holes (s_1, s_2). Furthermore, in the last three cases, the human will usually identify the hole with its host object so that, for instance, she would simply talk about "passing a sphere through the ring" and not through the "ring hole." This apparently subtle distinction may help to drastically simplify the problem representation. The spatial possibilities of objects with multiple holes and their possible interactions by passing through other objects may be interesting, but are not relevant for the problem. Thus, a first important simplification we make is to identify each *hole* with a *single-holed object*. The post hole seems to be an exception in this sense, as there is some difference between the ring being in the post down the hole or not. As we will see later, this will be handled by "partitioning" the post into two imaginary pieces: the hole itself (ph) and a connected long object (the post body p). Let's describe now our hole representation in more detail.

In this work we assume that each hole has two poles representing the two sides of the hole. These poles subdivide the space local to the hole into two parts, named the *hole subspaces*. As shown in Figure 2.

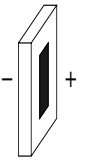


Fig. 2. Poles and subspaces of a hole

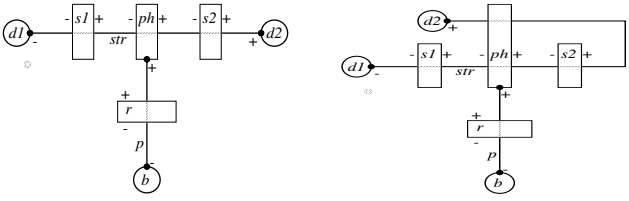
Figure 2 represent the hole poles and their relative subspaces. In this figure the hole is the shaded region, the poles and subspaces are represented by a '+' and a '-' sign. Therefore, the holes function as local (bi-dimensional) reference frames, as we can localise objects that are just in "front" or just "behind" any particular hole (but not sidewise objects). It is worth pointing out that, in this work, the holes have no dimension, i.e. the space *inside* the hole is null. This simplifying assumption can be easily dropped by considering a third symbol ('0' for instance) representing the spatial region that lies in between the two hole poles. In this paper, however, for each hole h we represent its corresponding poles as h^- and h^+. Furthermore, if a is a hole pole, then $-a$ represents the opposite one, so that $-h^- = h^+$ and $-h^+ = h^-$.

Apart from the hole sort, there are two objects in the puzzle that also share some common features: the string (str) and the post (p). If we momentarily forget the hole in the post, both objects are "long" in the sense that, in principle, they could be simultaneously crossing several holes. Another common feature is that we can recognise two tips in each of these objects: for instance, we would probably talk about the "right end of the string," or the "top end of the post," to put a pair of examples. Thus, we will consider a second sort called *long object*, in which we include p and str. For each long object l we define two tips l^- and l^+, following an analogous notation to that used wrt hole poles. Each tip of a long object can be *linked* to something else. For instance, the string tips are connected to the disks, whereas the bottom of the post is linked to the post base. Although encoded in the same sort, the string and the post have an obvious difference: the flexibility inherent in the former, which is not a characteristic of the latter. As we shall see, in this work the string's flexibility is reflected in the constraints imposed on the movements of the domain objects connected to it.

The remainder objects of the puzzle, that are the disks (d_1, d_2) and the post base (b), will just be classified as *regular* objects, without showing any particular feature, except perhaps that, due to their size, they can or cannot pass through each existing hole.

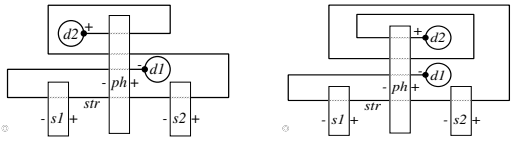
We illustrate the formalisation of the puzzle domain using diagrams. In these diagrams a box represents a *hole*, a circle a *regular object*, a thick line stands for a *long object* and a small black circle represents a link or connection. An example of this graphical representation is shown in Figure 3. Note how the post has been divided into a post hole (ph) linked to the top part of the post body (p). It may be reasonably objected that this division is not so natural, but it is also true that it would be possible to build an equivalent puzzle where, for instance, ph was a ring and p a second string.

Since each long object X can be crossing several holes, we will represent this using a list of crossings, called $chain(X)$. This list should collect the sequence of all hole crossings made by object X starting, for instance, from its negative tip and moving towards



(a) S_0 (initial state) (b) S_1 (d_2 passes left)

Fig. 3. A pair of puzzle states



(a) (b)

Fig. 4. Two different states that could not be distinguished without crossing directions

its positive one, whereas the same hole may occur several times in the list. Furthermore, the *direction* in which the string crosses the hole is also relevant. To see why, assume we represent the situation for Figure 4(a) simply as $chain(str) = [ph, s_1, ph, s_2, ph, ph]$. Then, we could not distinguish Figure 4(a) from Figure 4(b).

Figure 4(b) clearly represents a substantially different situation wrt Figure 4(a): the disk d_2 is now to the *right* (or positive side) of the post hole ph. Instead, a more suitable representation of Figure 4(a) would be: $chain(str) = [ph^-, s_1^+, ph^+, s_2^+, ph^-, ph^-]$.

Using the formalisation of the puzzle in terms of the list *chain*, presented in this section, we are able to define the basic actions on domain objects, as introduced below.

3 Acting on Objects

In this section we define the two actions that implement the basic movements on the puzzle's objects: the action $pass\_o$ (passing an object through a hole) and the action $pass\_h$ (passing a hole through another hole).

3.1 Moving Object Endings: Action $pass\_o$

The action $pass\_o(A, B)$ represents passing a long object tip A through some hole towards the hole pole B. For example, the execution $pass\_o(str^+, ph^-)$ in the initial state S_0 leads to S_1 (both depicted in Figure 3) and corresponds to moving the positive ending of str (currently linked to disk d_2) to the left of the post hole. It is clear that the execution of $pass(X^+, B)$ (resp. $pass(X^-, B)$) may equally mean that we are adding or removing the hole from $chain(X)$ depending on the context. For instance, the movement described above, $pass\_o(str^+, ph^-)$ should add ph^- to $chain(str)$ leading

to the list $[s_1^+, ph^+, s_2^+, ph^-]$ in S_1, whereas performing $pass\_o(str^+, ph^+)$ in that state should return us to the initial situation S_0, removing ph^- from the list.

The possible effects of $pass\_o$ are depicted in Figure 5.

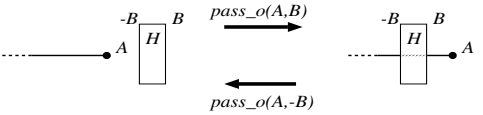


Fig. 5. Possible effects of $pass\_o$

Looking from the right to the left execution of $pass\_o$, we can conclude that, when we are performing $pass\_o(X^+, B)$ on the $chain(X) = [\ldots, -B]$, we must remove $-B$ from the tail of this chain. The analogous case would be $pass\_o(X^-, B)$ with $chain(X) = [-B, \ldots]$ where we would remove $-B$ from the head of $chain(X)$. If none of the two previous cases occur, then $pass\_o(A, B)$ is actually *inserting* a new crossing in $chain(X)$. Thus, $pass\_o(X^+, B)$ adds crossing B in the tail of $chain(X)$ whereas $pass\_o(X^-, B)$ adds crossing $-B$ to the chain head.

3.2 Passing Holes Through Holes: Action $pass\_h$

The previous action is not enough for describing the solution of the problem, since it does not take into account the movement of passing an (object containing a single) hole through another hole. To understand why, let us assume that, given the initial situation depicted in Figure 3 we tried to move the ring up. This is equivalent to move the post hole ph down the ring, that is, to pass ph through r^-. So we would need an action such as $pass\_h(A, B)$ where A is now a hole and B a hole pole. Back to the example, we would execute the action $pass\_h(ph, r^-)$ on the initial situation leading to the resulting state depicted in Figure 6.

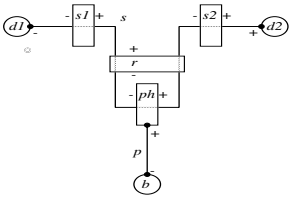
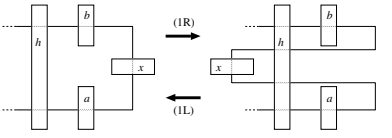


Fig. 6. Possible effect of $pass\_h$

The most relevant effect of this action is that the string chain, which was previously unrelated to the ring hole, has gained two new crossings as an effect of $pass\_h$. In other words, the list: $chain(X) = [s_1^+, \underline{ph^+}, s_2^+]$ has to be updated to: $chain(X) = [s_1^+, \underline{r^-, ph^+, r^+}, s_2^+]$.

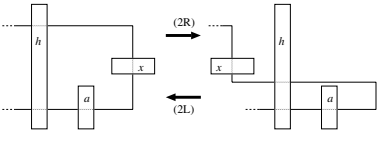
3.3 Possible Movements

This section presents some possible movements that can be operated applying the two rules defined above. In the diagrams, we assume that the uppermost and the rightmost tips of long objects are positive.



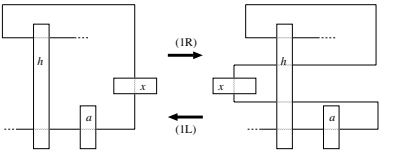
$[\ldots, h^+, a^+, \underline{x^+}, b^-, \ldots]$ $[\ldots, h^+, a^+, \underline{h^-, x^+, h^+}, b^-, \ldots]$

In Move (1R) we have that x is not contiguous to h in the chain. Therefore, by executing $pass\_h(x, h^-)$ we replace x^+ by the triple h^-, x^+, h^+. Movement (1L) starts in a state where x is preceded and succeeded by h in the chain but with alternate signs. A second possible movement would be:



$[\ldots, h^+, a^+, \underline{x^+, h^-}, \ldots]$ $[\ldots, h^+, a^+, \underline{h^-, x^+}, \ldots]$

The problem of Move 2R is that it cannot be applied when x is followed by h^+ instead of h^-, as shown in the instance of movement 1, as follows:



$[\ldots, h^+, a^+, \underline{x^+}, h^+, \ldots]$ $[\ldots, h^+, a^+, \underline{h^-, x^+, h^+}, h^+, \ldots]$

In general, for any hole e, assuming we want to execute $pass\_h(x, e)$ and x is crossed by some string, then for any string Y crossing x, and any occurrence of x in $chain(Y)$ we have the following list of possible movements:

(1R) $chain(Y) = [\ldots, a, \underline{x^z}, b, \ldots] \Longrightarrow [\ldots, a, \underline{e, x^z, -e}, b, \ldots]$ with $a, b \notin \{e, -e\}$ or $a = e, b = -e$.
(1L) $chain(Y) = [\ldots, \underline{-e, x^z, e}, \ldots] \Longrightarrow [\ldots, \underline{x^z}, \ldots]$
(2R) $chain(Y) = [\ldots, a, \underline{x^z, e}, \ldots] \Longrightarrow [\ldots, a, \underline{e, x^z}, \ldots]$ with $a \neq -e$
(2L) $chain(Y) = [\ldots, \underline{-e, x^z}, a, \ldots] \Longrightarrow [\ldots, \underline{x^z, -e}, a, \ldots]$ with $a \neq e\}$

state	$chain(p)$	$chain(str)$	next action(s)	movements
S_0	$[r^+]$	$[s_1^+, ph^+, s_2^+]$	$pass\_o(str^+, ph^-)$	
S_1	$[r^+]$	$[s_1^+, ph^+, s_2^+, ph^-]$	$pass\_o(p^+, r^-)$	(1R) $\times$ 2
			& $pass\_h(ph, r^-)$	
S_2	$[\,]$	$[s_1^+, r^-, ph^+, r^+, s_2^+, r^-, ph^-, r^+]$	$pass\_h(s_2, r^-)$	(1L)
S_3	$[\,]$	$[s_1^+, r^-, ph^+, s_2^+, ph^-, r^+]$	$pass\_h(r, ph^+)$	(2R)+(2L)
S_4	$[\,]$	$[s_1^+, ph^+, r^-, s_2^+, r^+, ph^-]$	$pass\_h(s_2, r^+)$	(1L)
S_5	$[\,]$	$[s_1^+, ph^+, s_2^+, ph^-]$		

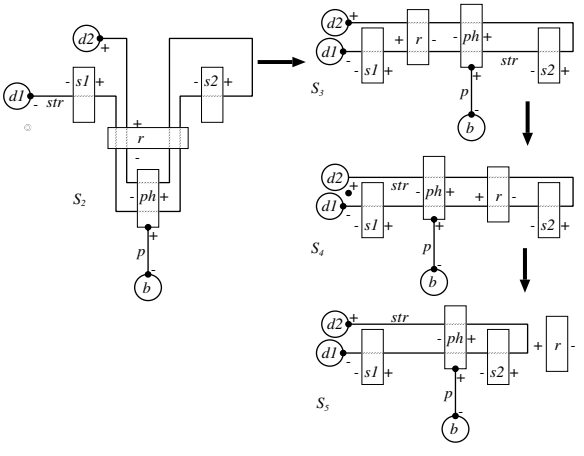


Fig. 7. A formal solution for the Fisherman's puzzle

Note that the above movements are complete, in the sense that if x occurs in $chain(Y)$ as follows $[\ldots, a, x^z, b, \ldots]$ both a and b could be equal to e, equal to $-e$ or none of the two. As a result, we would have $3 \times 3 = 9$ possibilities which, for reasons of space we do not depict here, but can be seen to be all covered by the movements above. The cases in which x is at head or tail end of the chain are also covered by assuming that the ends themselves are elements different from all the rest in the list.

Another important observation is that, while all the represented elements in each movement would be involved in the distinction of the movement type, only the underlined parts constitute the movement effect. This means, for instance, that in movement (2R), a is only used in the predecessor state, to establish that we have a (2R) movement and not a (1L), whereas in the successor state, it could be the case that a results moved to the left or even removed by the effect of another movement (remember that x may occur several times in the chain). An example of this *accumulated* movement would be, for instance, the execution of $pass\_h(x, h^+)$ on the list $[h^-, x^+, h^+, x^-, h^+]$ where we would perform (1L) on the first x and (2R) in the second leading to $[x^+, h^+, x^-]$.

With the representation developed above we can now formally express one solution to the Fisherman's puzzle and the sequence of states involved in its execution. Figure 7 shows this solution step by step and depicts the corresponding spatial configurations of states S_2 through S_5 (S_2 and S_3 were already shown in Figure 3).

Clearly, S_5 is a solution, since the ring r is not passed through any long object. Note that the action performed in state S_1 is actually a combined one. This is because moving the ending p^+ to r^- implies passing also the post hole, as p^+ and ph are linked.

The next section discusses an implementation of the puzzle into an action language.

4 A Simple Prolog Implementation

As an actions domain, our abstraction of the Fisherman's folly is quite simple in the sense that most complex features of actions reasoning are not required for the problem. We deal with two actions, $pass\_o$ and $pass\_h$ whose execution causes a direct effect on the fluents $chain(X)$, for each long object X. Rather than providing a precondition per each action, we have found more convenient to specify general constraints in which the actions are not executable. We have used a Prolog predicate *nonexecutable(S,A)* to represent when an action *A* cannot be performed in a state *S*, including the rules:

$$\begin{aligned}
&nonexecutable(_, pass_o([X,_],[H,_])) :- \\
&\qquad cannot_pass(X,H), !. \\
&nonexecutable(S, pass_o(P,[H,_])) :- \\
&\qquad member(linked_to(P) = X, S), cannot_pass(X,H), !. \\
&nonexecutable(_, pass_h(X,[H,_])) :- cannot_pass(X,H), !. \\
&nonexecutable(S, pass_h(X,_)) :- \\
&\qquad member(linked_to(_) = X, S), !.
\end{aligned}$$

The pairs *[X,Y]* are used to represent tips, so that for instance, *[p,+]* would stand for p^+. The fourth non-executability condition is used to force that, when an object tip is connected to a hole, the planner tries first to pass the object's tip and later the hole in a same transition. In this way we avoid irrelevant solutions where we can try to do it in the opposite ordering, obtaining exactly the same effects.

Of course, the main difficulty of this scenario from the standpoint of planning representation languages (STRIPS, ADL, PDDL) or even formalisms for action reasoning is the need for dealing with lists and pattern matching. In fact, this has motivated the choice of Prolog in order to build this preliminary prototype. Our implementation includes a Prolog predicate *process_chain(X,HP,L1,L2)* to describe the effect of performing $pass\_h(X, HP)$ on the chain list *L1* leading to the list *L2*. An example showing the implementation of movement (1R) is shown below.

$$\begin{aligned}
&process_chain(X, HP, [A,[X,S],B|Cs], Ds) :- \\
&\qquad opposite(HP, HP1), A \setminus = HP1, B \setminus = HP, !, \\
&\qquad process_chain(X, HP, [B|Cs], Ds0), \\
&\qquad Ds = [A, HP, [X,S], HP1|Ds0].
\end{aligned}$$

Note how the right neighbour of *[X,S]*, the crossed tip *B*, is used to keep processing the rest of the chain in the recursive call, and how the result of this recursive call *Ds0* may not contain *B* any more – it could be deleted by an accumulated movement (1L).

From the planning algorithm point of view, we have just implemented a blind search, relying on depth-first forward chaining with an iterative deepening strategy. Since the

plan is really short, the Prolog program[1] just takes 10.30 seconds to find a first solution, despite of the inefficient planning strategy.

It is interesting to observe that the program actually finds several solutions in five steps. For instance, apart from the obvious symmetric solution where we begin working with d_1 instead of d_2 making $pass(s^-, ph^-)$, we also get a variant of the depicted solution in Figure 7 where to reach state S_3 we execute instead the sequence $pass\_o(str^+, ph^-)$, $pass\_h(s2, r^-)$ and $pass\_o(p^+, r^-)$ & $pass\_h(ph, r^-)$. This solution is not valid for the original puzzle since, although both the sphere and the post can pass through the ring, they cannot do so *simultaneously*. For immediate future work, we plan extending our representation so that the predicate *cannot_pass* describes when a group of objects cannot be altogether simultaneously passing through a given hole.

We have also made some small variations of the original puzzle by changing some of the premises. For instance, by allowing spheres to pass through the post hole we directly get a shorter solution: $pass\_o(str^+, ph^-)$, then $pass\_h(s2, ph^-)$, that gets the string-disks-spheres tandem out of the post and, finally, $pass\_o(p^+, r^-)$ & $pass\_h(ph, r^-)$ to get the ring free.

5 Concluding Remarks

In this work we investigated a challenging problem for spatial reasoning and common sense knowledge formalisation, namely, the problem of reasoning about spatial domains that contain non-trivial objects such as holes and strings. We propose a representation whereby holes identify sub-spaces in which objects could be located. The string in this paper is formalised as a long object that restricts the movement of the objects linked to it. In fact, the flexibility of this object is not fully explored in this work, as we abstracted away the possibility of tying knots. This issue shall be investigated in future work.

The formalisation of spatial knowledge is not the only challenge in the domain assumed in this work. Solving the puzzle also involves interesting issues that are beyond search (or planning) through a state space. For instance, when changing the spatial configuration of the puzzle, a person has a *selective observation* of domain objects, whereby only a portion of the space is observed. Actions are, thus, only applied within this limited view of the scene. A second issue is the minimisation of the *spatial configuration complexity*; the string allows for the application of a sequence of actions rolling the string around the post, or through it, many times. Minimisation of the puzzle's configuration complexity could be used as an heuristic in an automated problem solver. However, how this complexity should be measured is still an open problem. Finally, when trying to solve the puzzle for the first time, a human agent may not know every constraint or every possible movement of the domain objects. Searching for an automated way in which such spatial knowledge can be assimilated is also a very challenging issue for further investigations.

[1] We have used SWI-Prolog 5.2.11 interpreter running on Linux Mandrake 10 on a Pentium IV 1.5 GHz with a RAM of 512 MB.

References

1. E. Clementini, P. di Felice, and D. Hernandez. Qualitative representation of positional information. *Artificial Intelligence*, 1997.
2. A. G. Cohn and S. M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-2):1–29, 2001.
3. D. Hernandez, E. Clementini, and P. di Felice. Qualitative distances. In W. Kuhn and A. Frank, editors, *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1995.
4. R. C. Meathrel and A. P. Galton. A hierarchy of boundary-based shape descriptors. In *Proc. of IJCAI*, pages 1359 – 1369, 2001.
5. T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi. Knot planning from observation. In *Proc. of the IEEE ICRA*, pages 3887 – 3892, 2003.
6. D. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In *Proc. of KR*, pages 165 –176, 1992.
7. K. Reidemeister. *Knot Theory*. BCS Associates, 1983.
8. P. Santos and M. Shanahan. Hypothesising object relations from image transitions. In *Proc. of ECAI-02)*, pages 292 – 296. 2002.
9. M. Shanahan. Default reasoning about spatial occupancy. *Artificial Intelligence*, 74(1): 147 – 163, 1995.
10. O. Stock, editor. *Spatial and Temporal Reasoning*. Kluwer Academic, 1997.
11. A. C. Varzi. Reasoning about space: The hole story. *Logic and Logical Philosophy*, 4:3 – 39, 1996.

A Causal Perspective to Qualitative Spatial Reasoning in the Situation Calculus

Mehul Bhatt[1], Wenny Rahayu[1], and Gerald Sterling[2]

[1] Department of Computer Science
La Trobe University Melbourne, Australia 3086
Tel.: +61-3-94791280
{mbhatt, wenny}@cs.latrobe.edu.au
[2] Air-Operations Division
Defence Science Technology Organisation
P.O. Box 4331 Melbourne Australia 3001
Tel.: +61-3-96267728
Gerald.Sterling@dsto.defence.gov.au

Abstract. We propose the utilisation of a general formalism to reason about action & change for reasoning about the dynamic purpose-directed aspects of spatial change. Such an approach is necessary toward the general integration of qualitative spatial reasoning with reasoning about the teleological aspects of spatial change. With this as the overall context, the main contribution of this paper is to illustrate first ideas relevant to providing a causal perspective to qualitative spatial reasoning using the situation calculus. With minimal notions about space & spatial dynamics, we perform a naive characterisation of objects based on their physical properties and investigate the key representational aspects of a topological theory of space, namely the region connection calculus, in the situation calculus. Further, ontological distinctions are made between various occurrents, i.e., actions and internal & external events, and a domain level characterisation of spatial occurrents in terms of their spatial pre-conditions & effects is performed so as to provide a causal perspective to spatial reasoning.

1 Introduction

Qualitative spatial reasoning (QSR) is an abstraction from precision oriented quantitative reasoning about the physical world [1]. Most research in qualitative spatial reasoning has focused on particular aspects of space such as topology [12], orientation [5], distance [7] etc and their integration thereof. However, relatively little work has explicitly addressed the need to account for the dynamic teleological or purpose-directed aspects of spatial change within a unified setup. Whereas qualitative spatial reasoning is concerned with the manner in which a set of spatial relationships evolve during a certain time interval, reasoning about the teleological aspects of a system involves reasoning about action and encompasses the goal directed aspects of spatial change. E.g., consider a travelling task from location L_1 to L_2 – Minimally, there are two main closely related aspects to this problem: (a) **Spatial**: The specific sequence of spatial transformations needed in order to achieve a certain desired configuration as well as its *legality*

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 430–440, 2006.

or consistency with regard to a set of spatial dynamics. (b) **Causal**: The overall goal or the telic aspect of achieving a desired spatial configuration that dictates why the agent wants to move from L_1 to L_2, which is orthogonal to how precisely to reach location L_2. Our research is driven by the need to treat inferences about the spatial aspects in an integrated manner with inferences about the causal aspects of a system; an endeavour which can be achieved via explicitly relating both the aspects, viz - the effects of actions/occurrences with a set of spatial dynamics. For instance, certain spatial transformations (or their sequences depending on the granularity) could be characterised as being the necessary criteria (pre-condition) for the happening of a particular occurrence. With this as our overall context, the main aim in this paper is to present first ideas on integrating qualitative spatial reasoning with reasoning about the dynamic goal directed & causal aspects of spatial change, an endeavour that can be achieved by formulating a set of causal axioms that relate domain specific *spatial occurrents* in terms of the spatial changes they represent. By spatial occurrents, we refer to events or actions that involve some form of transformation over the spatial structures being modelled. These occurrences can be dealt with at various levels of abstraction, from the most primitive & domain independent to complex, high-level aggregates that are built using the primitives. E.g., in a spatial theory relying solely on mereo-topological relations, a limited notion of occurrences can be defined by exploiting the concept of direct & continuous change between the relations. This abstraction has been used in [6] toward an event based qualitative simulation system. Similarly, a high-level approach would directly model domain specific occurrences on the basis of primitive spatial transitions; e.g., in [3], complex turn actions (such as *turn-left*) have been defined using primitive orientation relations. Our approach is to start with minimal notions about the nature of space & spatial dynamics and investigate the key representational issues that arise whilst dealing with these concepts in conjunction with actions, events & their effects in the context of a general formalism to reason about action & change. Specifically, we perform a naive characterisation of objects based on their physical properties into rigid & non-rigid types. Furthermore, we only use topological knowledge as a spatial metaphor with the continuity [5] of such relations being used to represent spatial dynamics. This approach, we hypothesize, is a necessary pre-requisite toward a general *spatio-teleological theory* that can integrate qualitative spatial reasoning (encompassing multiple aspects of space) with the teleological aspects of spatial change. Such a theory can be used as a basis for modelling dynamic phenomena in a variety of application contexts that require a useful predictive & explanatory capability.

2 Preliminaries - Notation and the Situation Calculus Formalism

The situation calculus as a representational tool for modelling dynamically changing worlds was first introduced in [11]. Some notation follows before the situation calculus formalism used in this paper is presented: Θ_S denotes the set of all spatial theory specific occurrents (spatial transitions). Θ_D denotes the set of all domain specific occurrences including actions (A) and internal (I) & external events (E). Θ denotes the set of all occurrences in the theory. By Φ, we denote the set of all functional & propositional fluents in the dynamic system being modelled. A similar distinction is also made between domain specific (Φ_D) & spatial theory specific (Φ_S) fluents. Finally, we assume

that there is a finite number of named occurrences and fluents. The situation calculus formalism used in this paper is essentially a first-order language with the following 5 classes of axioms: **(1)**. Domain specific *action pre-condition axioms* and possibility criteria for various *spatial transitions* definable in the spatial theory are specified using the binary predicate $Poss(\theta,\ s)$, where $\theta\ \in\ \{A\ \cup\ \Theta_S\}$. $Poss$ denotes that occurrence θ possible in situation s. **(2)**. The predicate $occurs(\theta,\ s)$ will be used to denote that all contextual conditions under which event θ can occur are satisfied in a situation s. Here, $\theta\ \in\ \{E\ \cup\ I\}$. **(3)**. A ternary predicate $Holds(\phi,\ v,\ s)$ denoting that fluent ϕ has the value v in situation s. For clarity, we will use it in the following alternative ways: **(a)**. $[\phi(s)\ =\ v]$ or **(b)**. $Holds(\phi,\ v,\ s)$), the latter essentially being the reified version. Finally, a non-determinate situation is expressed in the following manner:$[\phi(s)\ =\ \{\ v_1\ \vee\ v2\ \}]\ \equiv\ [Holds(\phi,\ v_1,\ s)\ \vee\ Holds(\phi,\ v_2,\ s)]$ **(4)**. The binary function $Result(\theta,\ s)$, which denotes the unique situation resulting from the happening of occurrence θ in situation s. Here, $\theta\ \in\ \{\Theta_D\ \cup\ \Theta_S\}$. **(5)**. A ternary $Caused(\phi,\ \gamma,\ s)$ predicate, where $\phi\ \in\ \Phi$, denoting that the fluent ϕ is *caused* to take on the value γ in situation s. Note that the $Caused$ predicate is always a direct (direct effects) or indirect link (indirect effects) between fluents & occurrences, an interpretation similar to its proposed use in [9]. For the predicate $Caused$, we need (1a) denoting that if a fluent ϕ is $Caused$ to take the value v in situation s, then ϕ holds the value v in s. (1b) & (2a-2b) denote the unique names & domain closure axioms for truth values & occurrences respectively. We assume that similar axioms for fluents are present as well. We also include a generic *frame axiom* (1c) thereby incorporating the principle of inertia or the non-effects of actions, viz - unless caused otherwise, a fluent's value will necessarily persist.

$$Caused(\phi,\ v,\ s)\ \supset\ Holds(\phi,\ v,\ s) \tag{1a}$$

$$true\ \neq\ false\ \wedge\ (\forall v)\,[v\ =\ true\ \vee\ v\ =\ false] \tag{1b}$$

$$\begin{aligned} Occurs(\theta,\ s)\ \vee\ Poss(\theta,\ s)\ \supset\ \{\neg(\exists v^{'})\,Caused(\phi,\ v^{'},\ Result(\theta,\ s))\ \supset \\ Holds(\phi,\ v,\ Result(\theta,\ s))\ \equiv\ Holds(\phi,\ v,\ s)\} \end{aligned} \tag{1c}$$

$$[\theta_i(\boldsymbol{x})\ \neq\ \theta_j(\boldsymbol{y})],\quad where\ i\ and\ j\ are\ different \tag{2a}$$

$$[\theta_i(\boldsymbol{x})\ =\ \theta_j(\boldsymbol{y})]\ \supset\ [\boldsymbol{x}\ =\ \boldsymbol{y}] \tag{2b}$$

3 Actions, Events and Spatial Reasoning - Ontological Issues

3.1 Naive Characterisation of Object Properties

An ontological distinction is made between an object and its *spatial extension*, the latter being denoted by the transfer function $space(object)$. We assume that the spatial extensions are regular regions of space that approximate the object in question. For representational clarity, we define the object-region equivalence axiom in (3a) and simply refer to spatial relationships as holding between *objects* of the domain. In conjunction with a theory of physical objects, the transfer function can be used to make the necessary distinctions. Objects may have varying properties at different times – e.g., take the case of a container filled with water. In this state, one may still drop a metal ball in the water/container. Now assume that in a later situation, the water is frozen and stays that

way for eternity. Such changes (e.g., water assuming solid-state) must be reflected as a change of property from a fully-flexible to a rigid object. Another issue is that of the classification of objects into strictly rigid & non-rigid types – when dealing with material (rigid) objects, topological changes can be understood to be the result of motion instead of other possibilities like continuous deformation that are possible with non-rigid objects. However, such a coarse distinction into strictly rigid & non-rigid types is insufficient since many objects exhibit properties of both types. E.g., an object such as a *room* cannot *grow* or *shrink*, but it can *contain* other objects. Therefore, the *room* taken as a whole can neither be classified as being strictly rigid, thereby not allowing interpenetration, nor is it a fully flexible non-rigid object like a water body that can *grow, shrink* or change *shape*. We do not attempt an extensive classification of object categories and the kind of changes permissible therein; rather, only as many distinctions relevant to *growth*, *shrinkage* & *containment* that are necessary in the context of the our example scenario are made. Based on the object properties in (4) and the topological relation ships in the RCC system ($RCC8 \equiv \{dc,\ ec,\ po,\ eq,\ tpp,\ ntpp,\ tpp^{-1},\ ntpp^{-1}\}$),domain independent constraints (rigidity (4c), non-rigidity (4d) & semi-rigidity (4e)) can be specified in order to rule out invalid spatial configurations that should not be permitted.

$$(\forall o_1,\ o_2)(\forall s)\ Holds(top(o_1,\ o_2),\ t_{rcc8},\ s) \equiv [(\exists r_1, r_2)\ space(o_1) = r_1 \wedge space(o_2) = r_2 \wedge Holds(top(r_1,\ r_2),\ t_{rcc8},\ s)] \tag{3a}$$

$$rigid(o,\ s) \equiv [\neg\ allows\_containment(o,\ s) \wedge \neg\ can\_deform(o,\ s)] \tag{4a}$$

$$non\_rigid(o,\ s) \equiv [allows\_containment(o,\ s) \wedge can\_deform(o,\ s)] \tag{4b}$$

$$(\forall o,\ o^{'})(\forall s)\ rigid(o,\ s) \wedge rigid(o^{'},\ s) \supset [Holds(top(o,\ o^{'}),\ t_{rcc},\ s)] \quad where\ t_{rcc} \in \{dc,\ ec\} \tag{4c}$$

$$(\forall o,\ o^{'})(\forall s)\ non\_rigid(o,\ s) \wedge non\_rigid(o^{'},\ s) \supset [Holds(top(o,\ o^{'}),\ t_{rcc8},\ s)] \quad where\ t_{rcc8} \in \{dc,\ ec,\ po,\ eq,\ tpp,\ ntpp,\ tpp^{-1},\ ntpp^{-1}\} \tag{4d}$$

$$(\forall o,\ o^{'})(\forall s)\ rigid(o,\ s) \wedge non\_rigid(o^{'},\ s) \supset [Holds(top(o,\ o^{'}),\ t_{rcc},\ s)] \quad where\ t_{rcc} \in \{dc,\ ec,\ po,\ eq,\ tpp,\ ntpp\} \tag{4e}$$

3.2 Occurrences - Actions and Events

Ontological distinctions into actions & events only apply at the level of the domain. In the spatial theory, there is only one type of occurrence, viz - a qualitative change as governed by the continuity constraints of the relation space. Within the spatial theory, it is meaningless to ascribe a certain spatial transition as being an event or action; such distinctions demanding a higher level of abstraction. As such, the classification of occurrences into actions & events will only apply at the level of the domain with the present spatial theory dealing only with one type of occurrence, namely a primitive spatial transition definable in it. We represent the same using $tran(t,\ o_i,\ o_i)$ (5b), which denotes that o_i & o_j transition to the state of being in relation t. An action is agent-centric and has a non-deterministic will associated with it. Simply, all pre-conditions for

a given action may be satisfied and yet the agent may not perform the action. Actions have possibility conditions that are expressed using the $Poss(\theta,\ s)$ predicate. Contrarily, events occur when the criteria for their occurrence holds. The occurrence criteria for an event is governed by its occurrence axiom (5d) where $\gamma(\langle \phi_1, \ldots, \phi_n \rangle,\ s)$ is a situation-dependent first-order formula characterising the necessary & sufficient conditions under which the event will *occur*. With events, a further distinction into external & internal events is made. Events, external to the system, which occur without there being any information as to their occurrence or cause are referred to as *external events*[1]. Events that are internal to the theory and which have associated occurrence criteria are referred to as *internal events*. Internal events are deterministic, i.e., if their corresponding occurrence axiom holds, the event will necessarily occur.

$$\Phi_S \ \equiv\ \{top(o_i,\ o_j),\ allows\_containment(o),\ can\_deform(o)\} \tag{5a}$$

$$\Theta_S \ \equiv\ \{tran(t_1,\ o_i,\ o_j), \ldots,\ tran(t_n,\ o_i,\ o_j)\} \tag{5b}$$

$$\gamma(\langle \phi_1, \ldots, \phi_n \rangle,\ s) \ \equiv\ (\exists\, v_1, \ldots, v_n)[Holds(\phi_1,\ v_1,\ s)\ \wedge \ldots \wedge\ Holds(\phi_n,\ v_n,\ s)] \quad where\ \phi_i\ \in\ \Phi \tag{5c}$$

$$\gamma(\langle \phi_1, \ldots, \phi_n \rangle,\ s)\ \supset\ Occurs(\theta,\ s) \quad where\ \theta\ \in\ \Theta_D \tag{5d}$$

4 A Causal Perspective to QSR in the Situation Calculus

We develop a spatial theory that can be exploited by defining domain specific spatial occurrents in terms of the spatial transformations that are definable in it. An example *delivery scenario* involving the distribution of delivery *objects* to a predefined set of *way-stations* via a delivery *vehicle* will be used throughout. Note that further elaborations of the scenario and the spatial theory are necessary in order to make optimal choices wrt. domain specific parameters such as minimal distance, stops etc.

4.1 Qualitative Spatial Reasoning with Topological Relations

Spatial change is modelled by time-varying topological relationships between objects (i.e., their spatial extensions). This is because topological distinctions are inherently qualitative in nature and they also represent one of the most general ways to characterise space. Precisely, we will use the Region Connection Calculus (RCC) [12] as the spatial theory and provide a causal perspective to it using the Situation Calculus [11]. Our justifications here being that it is necessary to view the problem at such a primitive level before an abstraction involving complex spatial occurrences is directly formalised using the situation (or other) calculus. The RCC-8 system being representative of a general class of similar relational techniques (e.g., JEPD base relations, continuity principle, compositional inference & consistency) that are common in the QSR area – our results can be generalised to cover the class of relational calculi having similar semantics.

[1] Practically, external events can be accounted for in a dynamic planner/controller by continuous interfacing with the external world.

Static Aspect - Composition Table Theorems as State Constraints. Various aspects of the RCC system can be represented using state constraints. E.g., every theorem from the RCC-8 composition table can be modelled as a state constraint thereby resulting in 8×8 constraints of the form in (6a). However, ordinary state constraints pose problems by containing indirect effects in them. In the context of the situation calculus, [9, 10] illustrates the need to distinguish ordinary state constraints from indirect effect yielding or *ramification constraints* – When these are present, it is possible to infer new effect axioms from explicitly formulated effect axioms together with the ramification constraints, i.e., ramification constraints lead to '*unexplained changes*'. This is evident in (6a), where the topological relationships between $o_1,\ o_2$ & o_3 are inter-related and changes in the relationship between any two objects will have an indirect effect on the third one. As such, instead of the ordinary constraint form in (6a), we will use an explicit notion of causality [9] by utilizing the $Caused(\phi,\ \gamma,\ s)$ predicate for the specification of such ramification constraints (6b). Other aspects of the RCC system like JEPD base relations too can be expressed using state constraints. In general, we need a total of n state constraints to express the jointly-exhaustive property of a set of n base relations and $[n(n\ -\ 1)/2)]$ similar constraints to express their pair-wise disjointness. Additionally, other properties such as the symmetry & asymmetry of the base relations too can be expressed using state constraints. Symmetric relations from the RCC-8 set include $dc,\ ec,\ po,\ eq$ whereas $tpp,\ tpp^{-1},\ ntpp,\ ntpp^{-1}$ are asymmetric.

$$(\forall s)\ Holds(top(o_1, o_2),\ dc,\ s)\ \wedge\ Holds(top(o_2, o_3),\ ec,\ s)\ \supset\ Holds(top(o_1, o_3),\ dr,\ s) \\ \vee\ Holds(top(o_1,\ o_3),\ po,\ s)\ \vee\ Holds(top(o_1,\ o_3),\ pp,\ s) \tag{6a}$$

$$(\forall s).\ [Holds(top(o_1,\ o_2),\ dc,\ s)\ \wedge\ Holds(top(o_2,\ o_3),\ ec,\ s)]\ \supset \\ Caused(top(o_1,\ o_3),\ \gamma,\ s) \quad where\, \gamma\ \equiv\ [dr\ \vee\ po\ \vee\ pp] \tag{6b}$$

Dynamic Aspect - Conceptual Neighbourhood Graph as Transitions. In the present spatial theory, the most primitive means of change is an explicit change of topological relationship between 2 objects (their spatial extensions). Let $tran(t_{rcc},\ o_i,\ o_j)$ (5b) denote such a change; read as, o_i and o_j *transition* to a state of being t_{rcc}. (7a) represents the (general) possibility axiom for the said transition. The binary predicate $neighbour(t_{rcc},\ t^{'}_{rcc})$ in (7a) is used to express the consistency of a direct continuous perturbation between 2 relations and is based on the conceptual neighbourhood principle [4], viz - relations t & $t^{'}$ are conceptual neighbours if 2 objects related by t can directly transition to the state of being $t^{'}$ & vice-versa. The conceptual neighbourhood graph for RCC-8 can be used to define a total of 8 transition axioms as in (7a).

Successor State Axioms - Causal Laws of the Spatial Theory. Successor state axioms (SSA) specify the causal laws of the domain – what changes as a result of various occurrences in the system being modelled. Generally, the SSA is based on a *completeness assumption* which essentially means that all possible ways in which the set of fluents may change is explicitly formulated, i.e., there are no indirect effects [13]. The SSA in this section is the one that is derived in the presence of ramification constraints. Recall our use of the ternary causal relation $Caused(\phi,\ \gamma,\ s)$ in (6b) toward the representation of the composition table theorems as state constraints. Precise details notwithstanding, what remains to be done is to minimize the causal relation by circumscribing

it (or using some other form of minimization) with the following set of axioms fixed – the foundational (1a-1c) & unique names axioms (2a-2b), the ramification constraints of the form in (6b) and the transition pre-conditions of the form in (7a). The result of minimization is the the *Causation Axiom* in (7b).

$$Poss(tran(t_{rcc},\ o_i,\ o_j),\ s) \equiv [\{space(o_i) = r_i \wedge space(o_j) = r_j\} \wedge \{(\exists\ t^{'}_{rcc})\ Holds(top(\ r_i,\ r_j),\ t^{'}_{rcc},\ s) \wedge neighbour(t_{rcc},\ t^{'}_{rcc})\}] \tag{7a}$$

$$Caused(top(o_i, o_k),\ t_{rcc8},\ s) \equiv \{t_{rcc8} = t_k \wedge (\exists o_j)[Holds(top(o_i,\ o_j),\ t_i,\ s) \wedge Holds(top(o_j,\ o_k),\ t_j,\ s)]\} \tag{7b}$$

$$\begin{aligned} Poss(e, s) \supset [&Holds(top(o_i,\ o_j),\ t_{rcc8},\ Result(e,\ s)) \equiv \\ &\{(\forall\ t^{'}_{rcc8})\ top(o_i,\ o_j,\ s) = t_{rcc8} \wedge e \neq tran(t^{'}_{rcc8},\ o_i,\ o_j)\} \vee \\ &\{e = tran(t_{rcc8},\ o_i,\ o_j)\} \vee \\ &\{(\exists o_k, t_i,\ t_j)\ Holds(top(o_i,\ o_k),\ t_i,\ s) \wedge Holds(top(o_k,\ o_j),\ t_j,\ s)\}] \end{aligned} \tag{7c}$$

The causation axiom in (7b) must be integrated with a Pseudo-SSA (PSA) (PSA is SSA without indirect effects) to derive the SSA in (7c). This final result is the SSA-Proper in (7c). In the SSA in (7c), we assume that the final disjunct is the only indirect effect, i.e., we only account for one theorem from the composition table, that too without stating it precisely. In practice, a complete axiomatisation accounting for all disjunctive labels within every non-deterministic composition table entry is required.

$$(\forall s)\ rigid(delivery\_agent,\ s) \tag{8a}$$

$$(\forall d)\ (\forall s)\ rigid(d,\ s) \quad where\ d \in D \tag{8b}$$

$$(\forall s)\ allows\_containment(vehicle,\ s) \wedge \neg\ can\_deform(vehicle,\ s) \tag{8c}$$

$$(\forall w)\ (\forall s)\ allows\_containment(w,\ s) \wedge \neg\ can\_deform(w,\ s),\ where\ w \in W \tag{8d}$$

$$(\forall w_1, w_2)(\forall s)\ [Holds(top(w_1,\ w_2),\ dc,\ s)] \tag{8e}$$

$$(\forall s)\ [\neg Holds(top(delivery\_agent,\ vehicle),\ ntpp,\ s)] \tag{8f}$$

4.2 Integrating Actions and Events with Spatial Reasoning

Domain Theory - Objects and Fluents. The scenario consists of a delivery *agent* & *vehicle*. D is the set of *delivery objects*; W is the set of pre-defined way-stations and O refers to the set of all objects in the domain. Note that ontologically, all elements in O are similar. Depending on their physical attributes, the objects have also been categorised (see 8a-8d) based on the constraints in (4). We also have domain specific constraints (see 8e-8f) specifying (a). way-stations are always *disconnected*, (b). the *delivery_agent* cannot be a *ntpp* of the *vehicle*. Such domain specific knowledge/constraints rule out invalid spatial configurations. Φ_D denotes the following domain specific fluents: $delivery(d,\ w_i,\ w_j)$ – there is a delivery (d) scheduled for pickup from way-station w_i and to be dropped-off at way-station w_j; $signalled(w)$ – a way-station (w)

has been signalled; $halted(w)$ – the delivery vehicle is presently halted at way-station w; and $moving(w_i,\ w_j)$ – the delivery vehicle in currently moving from way-station w_i to way-station w_j. We also have 1 functional fluent, namely $top(o_i,\ o_j)$ (topological relationship between o_i & o_j), with its denotations being closed under the set of RCC-8 base relations. Strictly speaking, this fluent is part of underlying spatial theory with its dynamics being governed by the causal laws defined for the spatial theory. With domain occurrences defined in terms of the spatial configuration of objects, $top(...)$ essentially functions as the bridge between the spatial & causal aspects of the modelled changes.

Domain Occurrences - Actions and Events. The occurrence of actions & events, which constitute the occurrences of the modelled dynamic system, are defined in terms of the spatial configurations which they either cause (effects) or depend on (pre-conditions). E.g., when the delivery vehicle (v) is a non-tangential part of the way-station (w), i.e., $Holds(top(v,\ w),\ s)$, the same is regarded as a occurrence of interest and a named *internal event*, viz - $arrival(w)$, is defined using the $occurs$ predicate in order to reflect the said event. Subsequent to the occurrence of the *arrival* event, the agent may decide to execute a control action (e.g., $halt$ at the way-station iff necessary); the main point here being that it is first necessary to be at the way-station (characterised by a particular spatial pattern) before the *halt* action can be performed.

$$A \equiv \{pickup\_delivery(d,\ w),\ droppoff\_delivery(d,\ w),\ halt(w),\ resume(w)\} \quad (10)$$

$$E \equiv \{schedule\_delivery(d,\ w_i,\ w_j)\}, \quad I \equiv \{signal\_pickup(w),\ arrival(w)\} \quad (11)$$

By Θ_D, we denote the set of all occurrences in the system, viz - actions and internal & external events. We have four actions (A) that the agent may execute. A brief description of these actions is as follows: *pickup_delivery(d, w)* represents the transfer of delivery object d into the way-station; *droppoff_delivery(d, w)* is the transfer of delivery object d from the delivery vehicle to way-station w; *halt(w)* is a control action that will stop the delivery vehicle whereas *resume(w)* is another control action that will resume the vehicle onto its onward course. The scenario has two internal events (I) that occur when certain contextual conditions are met; as explained before, $arrival(w)$ occurs when a spatial configuration denoting the presence of the vehicle inside the way-station is established. Similarly, *signal_pickup(w)* occurs at a way-station every time there is an object scheduled to be picked-up from that way-station. There is only one external event (E) in the system, namely *schedule_delivery* that occurs non-deterministically.

Action Pre-Condition, Occurrence and Effect Axioms. Action pre-conditions (12a) that must hold prior to actual execution are axiomatically specified using the binary predicate $Poss(\theta,\ s)$. Wherever possible, all action pre-conditions are defined in terms the spatial configuration of the domain objects (in addition to non-spatial criteria) that must hold for the action to be executable. Such an explicit link between an action & its spatial pre-requisites can be exploited whilst reasoning about the reachability of certain spatial configurations and the occurrences that either depend on them (pre-conditions) or directly cause them (effect axioms). Effect axioms, modelled using the ternary causal relation $Caused(\phi,\ \gamma,\ s)$, specify what changes directly as a result of the named occurrences. The effect axiom in (12b) denotes that if an occurrence θ either occurs or is possible in situation s, then the fluents $\phi_1, \ldots, \phi_n$ are *caused* to take on the values

$v_1, \ldots, v_n$ respectively. Whenever indirect effects are present, we assume that they are specified using ramification constraints (6b).

$$Poss(\theta,\ s) \equiv [Holds(\phi_1,\ v_1,\ s) \wedge \cdots \wedge Holds(\phi_n,\ v_n,\ s)] \tag{12a}$$

$$Occurs(\theta,\ s) \vee Poss(\theta,\ s) \supset [Caused(\phi_1,\ v_1,\ Result(\theta,\ s)) \wedge \cdots \wedge Caused(\phi_n,\ v_n,\ Result(\theta,\ s))] \tag{12b}$$

$$\begin{aligned} Occurs(\theta,\ s) \vee Poss(\theta,\ s) \supset [Holds(\phi,\ true,\ Result(\theta,\ s)) \equiv \\ \{\theta = (\theta_1^+ \vee \ldots \vee \theta_n^+)\} \vee \{Caused(\phi,\ true, s)\} \vee \\ \{Holds(\phi,\ true,\ s) \wedge \theta \neq (\theta_1^- \vee \ldots \vee \theta_n^-)\}] \end{aligned} \tag{12c}$$

Successor State Axioms - Causal Laws of the Domain. Unlike the SSA in (7c), the SSA for the domain theory (12c) universally quantifies over both events & actions. However, the semantics for (12c) are the same, i.e., 'As a result of occurrence θ, fluent ϕ holds in the resulting situation s *iff*–(a). θ explicitly causes ϕ to be $true$, or (b). ϕ is $caused$ to be $true$ as a ramification of something unknown, or (c). ϕ was originally $true$ and continues to be true on the basis of the principle of inertia (1c). This SSA is the reason why direct effects too have been specified using the $Caused$ predicate – On the basis of the effect axioms (12b), the ramification constraints (6b), frame axioms (1c) & minimization of effects (i.e, causal relation), causation axioms encompassing direct & indirect effects are obtained for the derivation of the SSA's [9, 10].

$$\Phi \equiv [Holds(top(o_1,\ o_2),\ t_1,\ s) \wedge Holds(top(o_2,\ o_3),\ t_2,\ s)] \tag{13a}$$

$$\begin{aligned} \Phi \wedge RCC8_{CT} \vdash \Phi^{'} \\ where\ \Phi^{'} \equiv [Holds(top(o_1,\ o_2),\ t_1,\ s) \wedge Holds(top(o_2,\ o_3),\ t_2,\ s) \wedge \\ Holds(top(o_1,\ o_3),\ t^{'},\ s)] \end{aligned} \tag{13b}$$

Initial Situation. A description of initial fluent values when no occurrences have happened is needed: For spatial fluents, there exist 2 classes: those which model the spatial relationship between objects (e.g., top) & those which characterise the dynamic object properties (e.g, $allows\_containment$). The case for non-spatial fluents (e.g., $moving$) and spatial fluents characterising object properties is trivial and will be omitted. As for the fluents encompassing spatial relationships, the initial situation description involving n domain objects requires a complete $n - clique$ description with $[n(n\ -1)/2]$ spatial relationships. However, relationships between some objects may be omitted in which case a complete initial situation description (with disjunctive labels) can be derived on the basis of the RCC-8 composition table. As an example, consider the simplest case involving 3 objects in (13): Φ denotes a partial description involving the 3 objects, viz - the relationship between objects o_1 and o_3 is unknown. Given Φ, $\Phi^{'}$ can be monotonically derived ($\vdash$) on the basis of Φ and the RCC-8 compositional table constraints (RCC_{CT})[2]. Here, Φ^1 is a monotonic extension of Φ in the sense that whilst new information is conjoined with Φ, none of the existing spatial knowledge is invalidated.

[2] Cui et al. [2] propose to maintain only those state descriptions that arise from pairs of objects in the initial situation. This will be the default behaviour when '$\vdash$' is not applied.

5 Discussion and Outlook

Our main interest lies in the development of a set of spatial primitives/patterns (and their combinations) that are necessary to satisfy a predictive & explanatory function for a particular domain. A corpus of spatial primitives (akin to a library of spatial dynamics) can be used to perform such functions within a domain; the main objective here being that it should be possible for domain modellers to use the underlying spatial theory for different application specific purposes. Specifically, the extensions can be used to model computational tasks such as: (a) *Spatial Planning*: Derive a sequence of spatial actions that will fulfill a desired objective. In other words, "How do we transform one spatial configuration into another?". Note that an online (incremental) plan generation approach that incorporates dynamically available information (e.g., an agent may have sensing capabilities) is more powerful in comparision to offline or a static one. As such, the theory developed in this paper can be implemented in the context of existing situation calculus based languages (e.g., [8]) that provide such capabilities. (b) *Causal Explanation*: Given a set temporally ordered snap-shots, the aim is to extract a explanatory causal model in terms of spatial occurrents from the given spatio-temporal data; the main objective here being, "What occurrences may have caused a particular spatial configuration, or a series of temporally ordered configurations?". Furthermore, it must be emphasized that explanation can only be provided for qualitative changes as reflected by the underlying relation space. For instance, in the absence of distance information, time varying distances between two *disconnected* objects will be not reflected within the theory. (c) *Spatial Projection & Interpolation*: Given an initial situation, predict all possible evolutions of the system or interpolate missing spatial knowledge between two temporal snap-shots of a changing system. Reachability of a particular situation could be characterised by the happening of one or more occurrences or by some spatial configuration of the objects in the domain.

References

[1] A. Cohn and S. Hazarika. Qualitative spatial representation and reasoning: an overview. *Fundam. Inf.*, 46(1-2):1–29, 2001. ISSN 0169-2968.

[2] Z. Cui, A. G. Cohn, and D. A. Randell. Qualitative simulation based on a logic of space and time. In *QR-92*, Heriot-Watt University, Scotland, 1992.

[3] F. Dylla and R. Moratz. Exploiting qualitative spatial neighborhoods in the situation calculus. In *Spatial Cognition*, pages 304–322, 2004.

[4] C. Freksa. Conceptual neighborhood and its role in temporal and spatial reasoning. *Decision Support Systems and Qualitative Reasoning*, pages 181–187. Amsterdam, 1991.

[5] C. Freksa. Using orientation information for qualitative spatial reasoning. In *Proceedings of the International Conference on GIS*, pages 162–178, London, 1992. Springer-Verlag.

[6] J. Gooday and A. Cohn. Transition-based qualitative simulation. In *Proceeding of the 10th International Workshop on Qualitative Reasoning*, pages 74–82. AAAI press, 1996.

[7] D. Hernandez, E. Clementini, and P. D. Felice. Qualitative distances. In *Spatial Information Theory: A Theoretical basis for GIS*, No. 988, pages 45–58. Lecture Notes in Computer Science, 1995.

[8] G. Lakemeyer and H. Grosskreutz. On-line execution of cc-golog plans. In *IJCAI*, pages 12–18, 2001.

[9] F. Lin. Embracing causality in specifying the indirect effects of actions. In *IJCAI*, pages 1985–1993, 1995.

[10] F. Lin and R. Reiter. State constraints revisited. *Journal of Logic and Computation*, 4: 655–678, 1994.

[11] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[12] D. A. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In *KR'92. Principles of Knowledge Representation and Reasoning*, Pages 165–176. Morgan Kaufmann, California, 1992.

[13] R. Reiter. The frame problem in situation the calculus: a simple solution (sometimes) and a completeness result for goal regression. pages 359–380, 1991.

PFORTE: Revising Probabilistic FOL Theories

Aline Paes, Kate Revoredo, Gerson Zaverucha, and Vitor Santos Costa

Department of Systems Engineering and Computer Science - COPPE
Federal University of Rio de Janeiro (UFRJ)
PO Box 68511, 21945-970, Rio de Janeiro, RJ, Brasil
{ampaes, kate, gerson, vitor}@cos.ufrj.br

Abstract. There has been significant recent progress in the integration of probabilistic reasoning with first order logic representations (SRL). So far, the learning algorithms developed for these models all learn from scratch, assuming an invariant background knowledge. As an alternative, theory revision techniques have been shown to perform well on a variety of machine learning problems. These techniques start from an approximate initial theory and apply modifications in places that performed badly in classification. In this work we describe the first revision system for SRL classification, PFORTE, which addresses two problems: all examples must be classified, and they must be classified well. PFORTE uses a two step-approach. The completeness component uses generalization operators to address failed proofs and the classification component addresses classification problems using generalization and specialization operators. Experimental results show significant benefits from using theory revision techniques compared to learning from scratch.

1 Introduction

The ability to represent individuals, their properties, and their relationships, makes first-order logic (FOL) a very expressive form of knowledge representation. On the other hand, FOL is limited by its inability to represent uncertainty. Recently, there has been great interest in integrating FOL based formalism with mechanisms for probabilistic reasoning, thus defining first-order probabilistic theories. Examples include: *Programming In Statistical Modeling (PRISM)* [Sato and Kameya, 1997], *Probabilistic Relational Models (PRM)* [Friedman et al., 1999], *Stochastic Logic Program (SLP)* [Muggleton, 2002], *Bayesian Logic Programs (BLP)* [Kersting and De Raedt, 2001], *Constraint Logic Programming (CLP($\mathcal{BN}$))* [Costa et al., 2003], *Markov Logic Networks (MLN)* [Richardson and Domingos, 2006] among others. The task of constructing such theories is often named *SRL* (Statistical Relational Learning).

Most SRL work assumes we want to learn from scratch. This may not be true. Often, one is supplied with a probabilistic first-order theory (PFOT). One assumes the theory to be approximately correct, i.e., only some points in its structure prevent it from correctly modelling the database. Arguably, it should be more efficient to search for such points in the theory and *revise* them, than to

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 441–450, 2006.

use an algorithm that learns a whole new theory from scratch. Prior work shows this approach to work well revising FOL theories [Richards and Mooney, 1995].

Learning or revising probabilistic first-order theories does introduce interesting novel issues. In SRL it is convenient to see examples as evidence for random variables. Thus, and differently from traditional Inductive Logic Programming (ILP), a SRL system should *model all examples*. The distribution of probabilities will then give an expectation for whether an example will take a specific value. In other words, every example should be satisfiable by the theory, and thus have a proof. Moreover, that proof should give us the best way to estimate the probability distribution. A theory should be revised if it does not cover examples, or if it does not generate the appropriate probability distribution.

Our problem can be stated as follows. Given an initial theory $T_0 = (BK, H_0)$, where BK is the background knowledge (invariant) and H_0 the part that can be modified, a set of examples E and a initial score of the theory $S_0 = M(E, T_0)$ according to some metric M, we want to discover a theory T with maximal score S according to M. In general, the problem does not have an exact solution, and we will therefore perform search according to some heuristic to find a theory T' that approaches T. For example, in the case of logical theories, FORTE [Richards and Mooney, 1995] refines theories by using two kinds of operators: *generalization* operators create more general theories, *specialization* operators create tighter theories.

One first step to approximate T is to apply generalization operators until all examples are covered [Revoredo and Zaverucha, 2002]. This is guaranteed to generate a theory consistent with the database. Unfortunately, such a theory may not maximize M. For example, the only theory that covers all examples may be the default theory. If so, the probability distribution over the examples will be given by the prior, and there would be no benefit in learning. A similar problem is well known in ILP: trying to fit all examples at all costs may lead to overfitting. This argues that the revision system should not only focus on covering all examples, but also on trying to find the best rules. Indeed, [Paes et al., 2005] suggests that even a limited heuristic such as applying generalization operators first, and specialization operators next can be useful in revising classification problems, although no detailed analysis was performed.

We present PFORTE, a revision system that performs general search over the revision points. Our strategy is as follows. First, PFORTE tries to find a good starting point by addressing theory incompleteness only, using generalizations operators. This phase stops when we cannot improve examples covering. Second, the generalized theory is the starting point for general search, using both generalization and specialization operators.

We present experimental results where PFORTE is used to revise an approximately correct theory and to learn a theory from scratch. Moreover, we show that the second step of PFORTE is indeed necessary to improve classification accuracy. The theories were constructed using BLPs [Kersting and De Raedt, 2001]. Our evaluations use three datasets and conditional log-likelihood as metric, as

several results suggest this metric to be very adequate for classification problems [Grossman and Domingos, 2004, Paes et al., 2005].

This paper is organized as follows: section 2 review some background knowledge; in section 3 our revision system PFORTE is presented; the experimental results are described in section 4; and finally some conclusions and future works are presented in section 5.

2 Preliminaries

In this section the way that the knowledge is represented in this work and first-order theory revision are briefly reviewed.

2.1 Knowledge Representation

Bayesian Networks are a popular knowledge representation technique [Pearl, 1988]. A Bayesian network is a direct acyclic graph where the nodes represent random variable that may be discrete or continuous. If there is an arrow from node X to node Y, X is said to be a parent of Y. Each node has a conditional probability distribution (CPD) that quantifies the effect of the parents on the node. An example for the brotherhood knowledge can be seen in Figure 1.

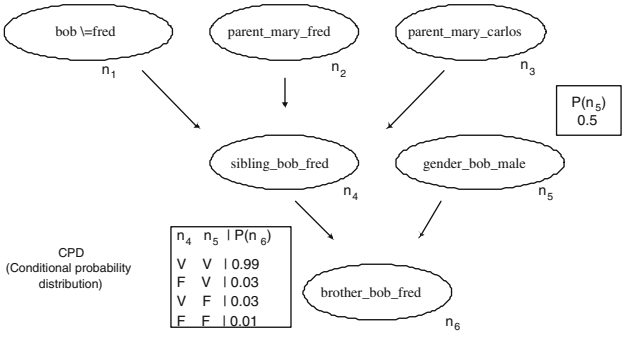


Fig. 1. Brotherhood knowledge represented through a Bayesian Network

Propositional clauses can also be used to represent the graph, as follows.

$brother\_bob\_fred \leftarrow gender\_bob\_male, sibling\_bob\_fred.$
$sibling\_bob\_fred \leftarrow parent\_mary\_bob, parent\_mary\_fred, bob \neq fred.$

In this representation, each atom corresponds to a node in the Bayesian network and the parents of the node become the antecedents of the clause.

Table 1. CPD associated to clause $brother(X,Y) \leftarrow gender(X,male), sibling(X,Y)$

gender(X)	sibling(X,Y)	P(brother(X,Y))
male	true	0.97
male	false	0.03
female	true	0.03
female	false	0.01

Bayesian networks are essentially a propositional representation: the set of variables is fixed and finite. Vastly more concise representations can be achieved using first-order knowledge representation, which commits to the existence of objects and relations among them and can express facts about some or all of the objects. For the brotherhood relationships the first-order representation is:

$brother(X,Y) \leftarrow gender(X,male), sibling(X,Y).$
$sibling(X,Y) \leftarrow parent(Z,X), parent(Z,Y), X \neq Y.$

In this work we use BLPs as our first-order probabilistic language. In this language, each ground atom in the program's model corresponds to a random variable, and each clause defines the ancestors for the ground instances of the head. Thus, the clauses above says that the random variables $brother(X,Y)$ depend on the random variables $gender(X)$ and $sibling(X,Y)$. The actual CPDs are given by an extra table, that associates each first-order clause with a CPD. As an example, Table 1 shows the CPD for clause, $brother(X,Y) \leftarrow gender(X)$, $sibling(X,Y)$.

2.2 Theory Revision

Theory refinement can be divided into two classes: theory revision and theory restructuring. Both aim at improving the quality of the theory. The revision task involves changing the answer set of the given theory, i.e., improving its inferential capabilities by adding previously missing answers, generalization or by removing incorrect answers, specialization [Wrobel, 1996].

A theory revision system receives an initial theory ($T_0 = (BK, H_0)$) and a set of examples ($E = E^+ \cap E^-$). This initial theory will include two components: an invariant component, named *background knowledge (BK)*, and one component that can be modified (H_0). When revising first order theories the set of examples is divided into positive (E^+) and negative examples (E^-). In this case, the revision process should generate a final theory ($T' = (BK, H')$) such that it will prove all the positive examples and none of the negative examples, thus the final theory will be consistent with the database. Learning in ILP can be seen as a particular case of theory revision where H_0 is initially empty.

In the case of a theory with multiple-predicate definitions many clauses can be involved in the proof of a negative example or in no proof for a positive example. Therefore, choosing the best operator is not immediate, and it becomes necessary

to find the theory´s points that need to be corrected. Depending on the type of example that is being considered we can define two types of revision points: *generalization revision points* which are the literals in a clause responsible for the failure of proving positive examples and other antecedents that may have contributed to this failure and *specialization revision points* which are the clauses used in successful proofs of negative examples. The specification of the revision point determines the type of revision operator that will be applied to make the theory consistent with the dataset.

Theory revision relies on operators that propose modifications at each revision point. Any operator used in first-order machine learning can be used. The most used ones are *delete-rule*, *add-antecedent* as *specialization operators* and *delete-antecedent*, *add-rule* as *generalization operators*. An example of revision system is FORTE [Richards and Mooney, 1995], which in each iteration it chooses the best revision to be implemented through an scoring function.

Several authors have shown that both Bayesian networks [Buntine, 1991], [Ramachandran and Mooney, 1998] and first-order theory revision systems [Wogulis and Pazzani, 1993], [Richards and Mooney, 1995] can learn more accurate theories with less data than purely inductive systems.

3 Probabilistic First-Order Theory Revision

Building on the contributions of FORTE we propose a probabilistic revision system called PFORTE which works in two steps. The first step tries to create a more general theory by using generalization operators only. The second step addresses classification problems using both generalization and specialization operators. The probabilistic theories we shall revise are BLPs. We chose conditional likelihood as our probabilistic scoring function.

An example in PFORTE is composed of a set of *instances*, which are the atoms we want to prove and then classify, a set of *facts* which will be used in the proof of the instances and a set of *evidences* that will be used in the classification of the instances. The schema of the dataset is as follows.

$$E = \{\{Instances, Facts, Evidences\}, ..., \{Instances, Facts, Evidences\}\}$$

Instances in an example are mutually dependent, but different examples are independent from each other.

As an example, consider the BLP in figure 2, where the CPD is omitted. PFORTE will first try to prove the instances. For each proven instance a Bayesian network is built using Knowledge-based Model Construction (KBMC) [Haddawy, 1999]. Then a support network [Kersting and De Raedt, 2002] is constructed for the example by the union of each provable instance's Bayesian network with each evidence's Bayesian network. The resultant support network built from the example in Figure 2 is the same as that one shown in Figure 1.

The support networks for all examples are used for learning the CPDs and to apply the probabilistic scoring function. Any Bayesian network inference algorithm can be applied in order to find each example's probabilities.

$E_1 = \{\{brother(bob, fred) = T, brother(bob, mary) = F\},$
$\{gender(bob), parent(mary, bob), parent(mary, fred)\},$
$\{parent(mary, fred) = T, sibling(bob, fred) = T, parent(mary, bob) = T\}\}$

$BLP : H = \{brother(X, Y)|gender(X), sibling(X, Y).$
$sibling(X, Y)|parent(Z, X), parent(Z, Y), X \neq Y.\}$
$KB = \{\ \}$

Fig. 2. Example E_1 and a BLP that will be used to build a Bayesian network from E_1

The first step of PFORTE focuses on generating a theory as complete as possible. To do so, PFORTE uses hill-climbing search to find all places where the BLP should be revised. We shall call these places *logical revision points*. These are points in the theory that failed in covering (proof) some example. Since there are only positive examples, the first step of PFORTE is restricted to generalization revision points, and only makes use of FORTE generalization operators. Each proposed modification is tested on the training set and receives a score. To calculate this score, PFORTE constructs the Bayesian networks from the examples, learns the CPDs, since the PFOT structure was modified, and calculates the probabilistic scoring function. From all generated revisions, PFORTE selects the one with the highest score. In case this best revision increases the logical covering of the validation set, it is implemented. PFORTE uses a validation set in order to avoid overfitting. This process continues until the first step cannot generate any revisions which logically improve the theory. It is expected that at the end of this process a PFOT consistent with the dataset is returned.

When learning the CPDs, the current structure is retained and the probability distributions which maximizes a given probabilistic scoring function, are searched. The algorithms considered in the literature, such as EM [Kersting and De Raedt, 2002, Koller and Pfeffer, 1997] and gradient ascent [Kersting and De Raedt, 2002], can be used for learning these CPD's.

The first step only addresses failed examples. It is possible to improve classification by modifying points that covered the examples, but did not classify them correctly. We call these points *probabilistic revision points*. Therefore, the second step tries to identify all probabilistic revision points in the consistent PFOT generated in the first step. To do so, the Bayesian networks are built from the instances, as explained above, and they are used to classify them. If the classification is different from the given one then the instance is considered incorrectly classified. The clauses that take part in that classification are considered as probabilistic revision points. Then, PFORTE tries to improve classification by applying either generalization or specializations operators to probabilistic revision points. This module also makes uses of FORTE operators, but when specializing rules antecedents can be added, if they improve the probabilistic scoring function value. These proposed modifications also receives scores, calculated in the same way as above. After PFORTE selects the revision with

the highest score, it verifies whether this revision can improve the probabilistic scoring function using the validation set. If an improvement is warranted, the best revision is implemented. This iterative hill-climbing process proceeds until no further revision improves probabilistically the PFOT. Even at this step, we enforce example covering when selecting operators. The algorithm is shown in Algorithm 1.

Algorithm 1. PFORTE Algorithm

```
repeat
   generate logical revision points;
   for each logical revision point
      generate possible revisions;
      for each revision
         learn the CPDs;
         calculate the score;
      update best revision found;
   if best revision improves the covering of the PFOT
      implement best revision;
until no revision improves the covering of the PFOT
repeat
   generate probabilistic revision points;
   for each probabilistic revision point
      generate possible revisions;
      for each revision
         learn the CPDs;
         calculate the score;
      update best revision found;
   if best revision improves the classification of the PFOT
      implement best revision;
until no revision improves the score of the PFOT
```

4 Experimental Results

This section presents experimental results of PFORTE showing the effectiveness of our approach in obtaining a revised PFOT in less time than when learning from scratch. Moreover, the results will point that the second step of PFORTE is essential to the revision system to return a PFOT with a better classification than if only the first step was applied.

All of our experiments use a Pentium4 3, 2GHz CPU based computer with 1GB of RAM. The EM algorithm was used to learn the CPDs, adapted as defined in [Kersting and De Raedt, 2002] and developed with the BNT toolbox [Murphy, 2001]. We used junction-trees for inference.

The first domain considered is Family domain [Quinlan, 1990]. This is a completely observed dataset with 245 instances in five examples. Family domain is a multi-predicate learning dataset with the eight concepts listed as

Table 2. Results after execution of 1° and 2° step in PFORTE, where COV means covering and ACC means accuracy

Domain	Initial COV	Initial ACC	Final COV	ACC after 1°step	ACC after 2°step
Family	79	64	100	84	95
Genetic	100	43	100	43	89
Bongard	0	0	100	55	83

follows: *wife, husband, mother, father, brother, sister, daughter and son.* The second domain is the Genetic, inspired by Friedman et al. [Friedman et al., 1999], holding 60 instances in 60 examples. Each example is relative to one person bloodtype. Finally, the last domain is the Bongard[1]. The dataset is composed of 60 examples where the concept we try to learn is "there is a triangle in a circle" [Kersting and De Raedt, 2002]. For the last two domains, we consider that 20% and 30% of all random variables are not observed, respectively.

The errors introduced in the initial theories were such as deleting rules, deleting antecedents, adding antecedents, changing an antecedent and changing a variable.

In order to avoid overfitting during training, we applied 5-fold cross validation approach to split the input data into disjoint training and test sets and, within that, a 5-fold cross-validation approach to split training data into disjoint training and tuning sets [Kohavi, 1995] similar to [Baião et al., 2003]. As the conditional log likelihood is a monotonic function we limited the size of clauses in four predicates. Restricting the maximum number of antecedents is often used in both ILP [Srinivasan, 2001] and Bayesian Networks [Grossman and Domingos, 2004].

Table 2 shows the results after the execution of the first and second steps of PFORTE. The measures presented are concerning covering (COV) (percentage of covered examples) and accuracy (ACC) (percentage of correctly classified examples). All of them are the average of measures for the five runs.

The covering for all revised domains is 100% showing that the revised theories are consistent with the datasets. Moreover, it is possible to see that the second step is essential since the accuracies are largely improved after its execution. The differences between accuracy after step one and after step two are statistically significant considering the significance level as $97,5\%$.

Table 3 shows a comparison between PFORTE learning from scratch and PFORTE revising the theories. In this table are presented the final accuracies and the average running time for both ways.

As it is possible to observe, PFORTE revising the theories is faster than PFORTE learning from scratch showing that it is much more efficient revise an approximate initial theory than learning it from scratch. Moreover, the accuracy are much better in two domains, considering $97,5\%$ as significance level. Therefore, to take in account an initial theory that can be modified is profitable.

[1] The Bongard problems, due to the Russian scientist M. Bongard, are well-known within ILP.

Table 3. Results after execution of PFORTE learning from scratch and PFORTE revising an initial theory

Domain	ACC_Revision	ACC_from_scratch	time_Revision(s)	time_from_scratch(s)
Family	95	80	8724	11906
Genetic	89	25	1656	2203
Bongard	83	83	2499	2743

5 Conclusion

In this paper we describe PFORTE, the first revision system for SRL classification. It works in two steps, the first one addressing theory incompleteness and the second one trying to improve classification. We compared experimentally PFORTE executing only the first step with the entire execution. It was possible to show that the second component is essential to obtain a good classifier.

Also, we compared experimentally the execution time and final accuracy of PFORTE learning from scratch and revising an initial theory. The experiments suggests that it may be much more efficient revising than learning from scratch and better accuracy can be produced as well, since more information about the initial theory is taking into account.

As we have shown, PFORTE can be used to learn from scratch. In this regard, PFORTE benefits from performing integrated search from the very beginning, and from localizing changes by only considering the revision points.

We believe our work presents a strong case for probabilistic theory revision. As future work, we intend to extend these results for other datasets. We have supported BLPs so far, but we believe our design can be applied to other SRL systems. Moreover, although we have based our work on FORTE, other approaches to theory revision can and should be considered.

Acknowledgements

The first author is partially financially supported by CAPES and the others by the Brazilian Research Council, CNPq.

References

[Baião et al., 2003] Baião, F., Mattoso, M., Shavlik, J., and Zaverucha, G. (2003). Applying theory revision to the design of distributed databases. In *Proc. 13th Int. Conf. on Inductive Logic Programming,LNAI 2835, Springer Verlag*, pages 57–74.

[Buntine, 1991] Buntine, W. (1991). Theory refinement on Bayesian networks. In *Proc. 17th Conf. Uncertainty in Artificial Intelligence*, pages 52–60.

[Costa et al., 2003] Costa, V., Page, D., Qazi, M., and Cussens., J. (2003). CLP(BN): Constraint logic programming for probabilistic knowledge. In *Proc. 19th Annual Conf. on Uncertainty in Artificial Intelligence*, pages 517–524.

[Friedman et al., 1999] Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. (1999). Learning probabilistic relational models. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence*, pages 1300–1309.

[Grossman and Domingos, 2004] Grossman, D. and Domingos, P. (2004). Learning bayesian network classifiers by maximizing conditional likelihood. In *Proc. 21th Int. Conf. on Machine Learning*, pages 361–368.

[Haddawy, 1999] Haddawy, P. (1999). An overview of some recent developments on bayesian problem solving techniques. *AI Magazine - Special issue on Uncertainty in AI*, 20(2):11–29.

[Kersting and De Raedt, 2001] Kersting, K. and De Raedt, L. (2001). Towards combining inductive logic programming with Bayesian networks. In *Proc. 12th Int. Conf. on Inductive Logic Programming, LNAI 2157 Springer Verlag*.

[Kersting and De Raedt, 2002] Kersting, K. and De Raedt, L. (2002). Basic Principles of Learning Bayesian Logic Programs. Technical Report 174, University of Freiburg.

[Kohavi, 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 1137–1145.

[Koller and Pfeffer, 1997] Koller, D. and Pfeffer, A. (1997). Learning probabilities for noisy first-order rules. In *Proc. 15th Int. Joint Conf. on Artficial Intelligence*, pages 1316–1323.

[Muggleton, 2002] Muggleton, S. (2002). Learning structure and parameters of stochastic logic programs. In *Proc. 12th Int. Conf. on Inductive Logic Prgramming,LNAI 2583, Springer Verlag*, pages 198–206.

[Murphy, 2001] Murphy, K. (2001). The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33.

[Paes et al., 2005] Paes, A., Revoredo, K., Zaverucha, G., and Costa, V. (2005). Probabilistic first-order theory revision from examples. In *Proc. 15th Int. Conf. on Inductive Logic Programming, LNCS 3625, Springer Verlag*, pages 295–311.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Inteligent Systems: networks of plausible inference*. Morgan Kaufman.

[Quinlan, 1990] Quinlan, J. (1990). Learning logical definitions from relations. *Machine Learning*, 5:239–266.

[Ramachandran and Mooney, 1998] Ramachandran, S. and Mooney, R. (1998). Theory refinement of bayesian networks with hidden variables. In *Proc. 15th Int. Conf. on Machine Learning*, pages 454–462.

[Revoredo and Zaverucha, 2002] Revoredo, K. and Zaverucha, G. (2002). Revision of first-order Bayesian classifiers. In *Proc. 12th Int. Conf. on Inductive Logic Programming, LNAI 2583,Springer Verlag*, pages 223–237.

[Richards and Mooney, 1995] Richards, B. L. and Mooney, R. J. (1995). Automated refinement of first-order Horn-clause domain theories. *Machine Learning*, 19:95–131.

[Richardson and Domingos, 2006] Richardson, M. and Domingos, P. (2006). Markov Logic Networks. *Machine Learning*, 62:107–136.

[Sato and Kameya, 1997] Sato, T. and Kameya, Y. (1997). Prism: A language for symbolic-statistical modeling. In *Proc. 15th Int. Joint Conf. on Artificial Intelligence*, pages 1330–1339.

[Srinivasan, 2001] Srinivasan, A. (2001). *The Aleph Manual*.

[Wogulis and Pazzani, 1993] Wogulis, J. and Pazzani, M. (1993). A methodology for evaluationg theory revision systems: results with Audrey II. In *Proc. 13th Int. Join Conf. on Artificial Intelligence*, pages 1128–1134.

[Wrobel, 1996] Wrobel, S. (1996). First-order theory refinement. In Raedt, L. D., editor, *Advances in Inductive Logic Programming*, pages 14–33. IOS Press.

Rule Schemata for Game Artificial Intelligence*

Flávio S. Corrêa da Silva[1] and Wamberto W. Vasconcelos[2]

[1] Dept of Computer Science, Univ de São Paulo, Brazil
fcs@ime.usp.br
[2] Dept of Computing Science, Univ of Aberdeen, UK
wvasconcelos@acm.org

Abstract. Rule-based systems are a promising means to specify interface standards for artificial intelligence tools and modules for games, as advocated by the International Game Developers Association. Rules, however, can be too flexible, allowing undisciplined and "dirty" programming styles and solutions. We advocate in this paper that although rules are a good starting point towards standardising artificial intelligence techniques in games, they must be complemented with automatically verifiable rule schemata to ensure the appropriate implementation of such techniques and theories. We illustrate our point with a specific rule-based implementation of a theory of norms for synthetic characters which enables the specification of sophisticated behaviours.

1 Introduction

Mutual cooperation and integration between Artificial Intelligence and Computer Games has been acknowledged as a desirable by their corresponding research and industrial communities. From the standpoint of Artificial Intelligence research, computer games are an ideal means for experimentation and presentation of research results. Games are also among the most successful areas of industrial application of artificial intelligence techniques [1]. From the standpoint of Computer Games, artificial intelligence has been hailed as the next breakthrough for gaming experience, after the "saturation" of computer graphics and visual realism [2,3,4].

Hence, a warm welcome must be given to the efforts of the International Game Developers Association (IGDA[1]) to propose to the game industry interface standards for artificial intelligence tools and modules for games, game engines and game editors. As presented in [5], the proposed standards for rule based systems shall be adapted from the Java Specification Request JSR 94 – Java Rule Engine API [6].

Rules are expressive, declarative, flexible, and can be executed very efficiently. These features rank rule-based inference engines high as a potential foundation on which to build artificial intelligence techniques and methods to control synthetic characters in computer games [2]. The problem with rules is that they

* Work partially sponsored by FAPESP and CNPq – Brazil.

[1] http://www.igda.org/

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 451–461, 2006.

can be *too* flexible. Although designed for declarative programming, rules – and more specifically rule based systems based on JSR 94 – give rise to all sorts of undisciplined and non-declarative constructs. We advocate that, although rules are a good starting point towards standardising the incorporation of artificial intelligence techniques to games, they must be complemented with automatically verifiable rule schemata that ensure the appropriate implementation of such techniques.

This is the central message of this article. In order to justify this, we present a specific proposal – namely, a rule-based implementation of a theory of norms of behaviour for synthetic characters. Although we expect this theory and its proposed implementation to be of interest in themselves for game developers and artificial intelligence researchers, we present it as a concrete case in which a rule-based schema – instead of just rules – enables the development of sophisticated behaviour for synthetic characters.

Let us consider the situation in which a synthetic character has a set of actions that can be performed at a particular point of the game – for instance, a warrior may have the choice of killing, maiming or freeing a prisoner. The decision procedure to choose (or rank) which actions to take should be independent of the design of how the characters interact. The set of permitted actions itself, however, is assembled taking into account the characters' interactions up to that point in the game. In our warrior example, if the prisoner in question in the past had killed the warrior's father, then the freeing option would be ruled out by the warrior's internal convictions, that is, the warrior would be prohibited by its own code of ethics from taking this action. In the next sections we expand what precisely we mean by *prohibitions* as well as the complementary notions of *permissions* and *obligations* and how these define computational behaviours.

This article is organised as follows: in Section 2 we introduce the foundations upon which we build our theory, based on deontic logics. In Section 3 we present the theory itself and show how it can be implemented as a rule schema, thus justifying our statement that rule schemata are useful to discipline the implementation of artificial intelligence theories. In Section 4 we review related work. Finally, in Section 5 we present some conclusions and proposed future work.

2 A Theory of Norms for Synthetic Characters

Social norms can be encoded in a multitude of ways. We are particularly interested in theories of norms that can be turned into computer programs to control the behaviour of synthetic characters, and from which we can guarantee mathematical properties to allow the verification of desirable features of norms, *e.g.*, whether the norms are consistent. As stated in section 1, we want to consider the dynamics of social norms, taking into account that permitted, obliged and prohibited actions are influenced by the characters' interactions.

A natural choice to express these norms is *deontic logics*. As presented in [7], deontic logics date as far back as medieval philosophy. The modern treatment of this family of logics, however, is commonly reputed to have started with Mally [8]

and von Wright [9]. Deontic logics add to classical (propositional) logics modal qualifiers corresponding to the notions of *permissions*, *prohibitions* and *obligations*. We present here a very brief account of a simple class of *monadic deontic logics*. Monadic deontic logics have monadic modal operators, *i.e.*, operators that apply to single propositions, *e.g.*, we can have a proposition p and a deontic sentence $\mathsf{O}p$, to be read as "p is obligatory". Our presentation is, more specifically, the so-called *Smiley-Hanson systems of monadic deontic logics* [10].

In our presentation, we enrich the language with predicates over finite models, making the language more adequate to express complex patterns of interactions. Strictly speaking, we do not add expressive power to a propositional language by adding predicates over finite models, but we make the language more "ergonomic": predicates are a shorthand for (lengthier) purely propositional sentences.

Definition 1. *The alphabet of our language consists of (1) A finite set of constants* $\mathcal{C} = \{c_1, \ldots, c_n\}$. *(2) A finite set* $\mathcal{P} = \{p_1^{m_1}, \ldots, p_s^{m_s}\}$ *of predicates* $p_j^{m_j}$ *of fixed arities* m_j*; this set includes the special 0-ary predicates* $\top$ *(*verum *or true) and* $\bot$ *(*falsum *or false). (3) A denumerable set of variables* $\mathcal{X} = \{X_1, X_2, \ldots\}$. *(4) The set of logical connectives* $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$. *(5) The set of deontic modalities* $\{\mathsf{O}, \mathsf{P}\}$, obligation *and* permission*, respectively.*

We do not explicitly add parentheses to our alphabet, but we shall use them whenever necessary to make the reading of sentences clearer.

Definition 2. *Our set of terms* $\mathcal{T}$ *is defined as any element* $t \in \mathcal{C} \cup \mathcal{X}$.

Definition 3. *The set of sentences* $\mathcal{S}$ *is the smallest set such that (1) For every k-ary predicate* $p^k \in \mathcal{P}$ *and k terms* $t_1, \ldots, t_k \in \mathcal{T}$*, then* $p^k(t_1, \ldots, t_k) \in \mathcal{S}$. *(2)* $\top, \bot \in \mathcal{S}$. *(3) If* $\varphi, \psi \in \mathcal{S}$*, then* $(\neg\varphi), (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi), (\varphi \leftrightarrow \psi) \in S$. *(4) If* $\varphi \in S$*, then* $\mathsf{O}\varphi, \mathsf{P}\varphi \in S$.

We add the abbreviation $\mathsf{F}\varphi \equiv \neg\mathsf{P}\varphi$ to capture the intuitive notion of prohibition: φ is prohibited iff it is not permitted.

Definition 4. *The sentences of the form* $p^k(t_1, \ldots, t_k)$ *are called atomic sentences.*

Definition 5. *The sentences of the form* $\mathsf{O}\varphi, \mathsf{P}\varphi$ *and* $\mathsf{F}\varphi$*, in which* φ *is an atomic sentence, are called simple deontic sentences.*

We assume implicitly that every variable occurring in a sentence is universally quantified. Thus, the following intuitive readings of the sentences below are assumed, in which $kills(t_1, t_2)$ is read as "t_1 kills t_2" and $father(t_1, t_2)$ is read as "t_1 is the father of t_2":

(1) $father(wizard, warrior)$: "*wizard*" is the father of "*warrior*".
(2) $kills(prisoner, wizard)$: "*prisoner*" kills "*wizard*".
(3) $kills(X_1, X_2) \wedge father(X_2, X_3) \rightarrow \mathsf{O}kills(X_3, X_1)$: for every X_1, X_2, X_3, if X_1 kills X_2 and X_2 is the father of X_3, then X_3 is obliged to kill X_1.
(4) $\mathsf{O}\, kills(warrior, prisoner)$: "*warrior*" is obliged to kill "*prisoner*".

Now one natural question to ask is: can we entail sentence **(4)** from sentences **(1)**, **(2)** and **(3)**? In order to do so, we need some additional definitions.

Definition 6. *A substitution σ is a finite, possibly empty set of pairs X/t, where $X \in \mathcal{X}$ is a variable and $t \in \mathcal{T}$ a term.*

Definition 7. *The application of σ to a construct $C \in \mathcal{S} \cup \mathcal{T}$ (a term or a sentence of our language), denoted by $C \cdot \sigma$, is defined as follows: (1) $c \cdot \sigma = c$ for every constant $c \in \mathcal{C}$ (2) $X \cdot \sigma = t \cdot \sigma$ if $X/t \in \sigma$; otherwise $X \cdot \sigma = X$ (3) $p(t_1, \ldots, p_n) \cdot \sigma = p(t_1 \cdot \sigma, \ldots, t_n \cdot \sigma)$ (4) $\top \cdot \sigma = \top, \bot \cdot \sigma = \bot$ (5) $(\neg\varphi) \cdot \sigma = \neg(\varphi \cdot \sigma), (\varphi \wedge \psi) \cdot \sigma = (\varphi \cdot \sigma \wedge \psi \cdot \sigma), (\varphi \vee \psi) \cdot \sigma = (\varphi \cdot \sigma \vee \psi \cdot \sigma), (\varphi \rightarrow \psi) \cdot \sigma = (\varphi \cdot \sigma \rightarrow \psi \cdot \sigma), (\varphi \leftrightarrow \psi) \cdot \sigma = (\varphi \cdot \sigma \leftrightarrow \psi \cdot \sigma)$ (6) $\mathsf{O}\varphi \cdot \sigma = \mathsf{O}(\varphi \cdot \sigma), \mathsf{P}\varphi \cdot \sigma = \mathsf{P}(\varphi \cdot \sigma)$*

Definition 8. *A substitution σ is a unification of two sentences φ and ψ iff $\varphi \cdot \sigma = \psi \cdot \sigma$; we can also say that σ unifies φ and ψ.*

Definition 9. *Our rules of inference are σ-Modus Ponens – $\frac{\varphi, \xi \rightarrow \psi}{\psi \cdot \sigma}(\sigma)$ for any substitution σ that unifies φ and ξ, that is, $\varphi \cdot \sigma = \xi \cdot \sigma$, and O-necessitation – $\frac{\varphi}{\mathsf{O}\varphi}$.*

A deontic logic is fully specified once we supply its *deontic relations*, namely how permissions, obligations and prohibitions relate to each other. The specification of deontic relations is a subtle and treacherous task that frequently leads to the formulation of conflicting theories (*e.g.*, a theory in which everything that is permitted is also prohibited).

The Smiley-Hanson systems of monadic deontic logics are interesting from a formal point of view, since they admit semantic models similar to those found in regular modal logics. The price we may pay to adopt these systems is to lose the capability of expressing some subtleties of deontic relations that can be important in specific domains. The Smiley-Hanson systems of monadic deontic logics are based on the following axiom schemata:

(1) all propositional tautologies	(2) $\mathsf{P}\varphi \rightarrow \neg\mathsf{O}\neg\varphi$	(3) $\mathsf{O}(\varphi \rightarrow \psi) \rightarrow (\mathsf{O}\varphi \rightarrow \mathsf{O}\psi)$
(4) $\mathsf{O}\varphi \rightarrow \mathsf{P}\varphi$	(5) $\mathsf{O}\varphi \rightarrow \mathsf{OO}\varphi$	(6) $\mathsf{PO}\varphi \rightarrow \mathsf{O}\varphi$
(7) $\mathsf{O}(\mathsf{O}\varphi \rightarrow \varphi)$	(8) $\mathsf{O}(\mathsf{PO}\varphi \rightarrow \varphi)$	

Each axiom schema admits an informal reading, some of which are more intuitive than others. Axiom schema (4), for example, states that every obligation is permitted. These axiom schemata are the building blocks to define specific logics. As presented in [10], we can build several different logics using subsets of these axiom schemata whose semantics are similar to those found in modal logics.

Definition 10. *A sentence φ is provable if, and only if, it is an axiom, or it can be derived from provable sentences using the inference rules of Def. 9 above.*

Definition 11. *A set of sentences Ψ is inconsistent if there are $\psi_1, \ldots, \psi_n \in \Psi, n \geq 1$ such that $(\psi_1 \wedge \ldots \wedge \psi_n) \rightarrow \bot$ is provable.*

Definition 12. *A sentence φ is provable from a set of sentences Ψ (denoted as $\Psi \vdash \varphi$) if the set $\Psi \cup \{\neg\varphi\}$ is inconsistent.*

Definition 13. *A model is a triple $\mathcal{U} = \langle \mathcal{W}, R, V \rangle$ where (1) $\mathcal{W}$ is a non-empty set of situations. (2) $R \subseteq \mathcal{W} \times \mathcal{W}$ is a co-permissibility relation, depicting which situations can be reached from each other. (3) V is a truth assignment, associating to each sentence φ a set $V(\varphi) \subseteq \mathcal{W}$, which is the set of situations in which φ holds.*

Intuitively, a situation can be considered as a scene of a game. For example, we could have the following situations: (1) $w_1 \in \mathcal{W}$: the warrior is surrounded by enemies. (2) $w_2 \in \mathcal{W}$: the warrior evades the enemies. (3) $w_3 \in \mathcal{W}$: the warrior is killed by its enemies. From situation w_1 we could move to situation w_2, situation w_3 or continue in w_1. This would amount to a co-permissibility relation R such that $(w_1, w_1), (w_1, w_2), (w_1, w_3) \in R$.

Definition 14. *A sentence is ground if it contains no variables. Given any sentence φ, we define $gr(\varphi)$ as the set of all ground sentences $\overline{\varphi}$ such that there is a substitution σ and $\varphi \cdot \sigma = \overline{\varphi}$.*

Definition 15. *Given a model $\mathcal{U} = \langle \mathcal{W}, R, V \rangle$ and $w \in \mathcal{W}$, we define truth conditions as follows, where $p(t_1, \ldots, t_n)$, $\overline{\varphi}$ and $\overline{\psi}$ are always ground:*

(1) $\models_w \varphi$ *iff* $\models_w \overline{\varphi}$ *for all* $\overline{\varphi} \in gr(\varphi)$.
(2) $\models_w p(t_1, \ldots, t_n)$ *iff* $w \in V(p(t_1, \ldots, t_n))$.
(3) $\models_w \top$.
(4) $\not\models_w \bot$.
(5) $\models_w \neg\overline{\varphi}$ *iff* $\not\models_w \overline{\varphi}$.
(6) $\models_w (\overline{\varphi} \wedge \overline{\psi})$ *iff* $\models_w \overline{\varphi}$ *and* $\models_w \overline{\psi}$.
(7) $\models_w (\overline{\varphi} \vee \overline{\psi})$ *iff* $\models_w \overline{\varphi}$ *or* $\models_w \overline{\psi}$.
(8) $\models_w (\overline{\varphi} \rightarrow \overline{\psi})$ *iff* $\models_w \neg\overline{\varphi}$ *or* $\models_w \overline{\varphi} \wedge \overline{\psi}$.
(9) $\models_w (\overline{\varphi} \leftrightarrow \overline{\psi})$ *iff* $\models_w (\overline{\varphi} \rightarrow \overline{\psi})$ *and* $\models_w (\overline{\psi} \rightarrow \overline{\varphi})$.
(10) $\models_w \mathsf{F}\overline{\varphi}$ *iff* $\models_w \neg\mathsf{P}\overline{\varphi}$
(11) $\models_w \mathsf{O}\overline{\varphi}$ *iff for every* $w', (w, w') \in R$ *then* $\models_{w'} \overline{\varphi}$.
(12) $\models_w \mathsf{P}\overline{\varphi}$ *iff for some* $w', (w, w') \in R$ *then* $\models_{w'} \overline{\varphi}$.

Definition 16. *Given a sentence φ and a set of sentences Ψ, (1) Ψ is not satisfiable if there is no model $\mathcal{U} = \langle \mathcal{W}, R, V \rangle$ and $w \in \mathcal{W}$ such that simultaneously $\models_w \psi$ for all $\psi \in \Psi$. (2) φ is semantically entailed by Ψ (denoted as $\Psi \models \varphi$) if $\Psi \cup \{\neg\varphi\}$ is not satisfiable.*

A model should be a characterisation of how permissions, prohibitions and obligations relate to each other in the world we are analysing (or designing, in the case of computer games). Observing that a co-permissibility relation determines a directed graph connecting situations, this characterisation can be built in terms of topological features that we may enforce in the model.

An important and remarkable feature of Smiley-Hanson systems of monadic deontic logics is that the proof theory presented above is sound and complete with respect to particular classes of models, characterised by topological features of their co-permissibility relations.

3 A Rule Schema for the Theory of Norms

We use the logics outlined in the previous section as the foundation for a computational theory of norms, used to control the behaviour of synthetic characters in computer games. The rule schema we present here is a variation of the work on norm-aware agent societies for e-commerce developed in [11,12,13]. As advanced in the previous sections, this theory is implemented as a rule schema, fully compatible with JSR 94. We restrict our language to the following

- $do(action_i(c, t_1, \ldots, t_k))$: represents an action $action_i$ to be performed by character c; the action can be parameterised by terms $t_1, \ldots, t_k$.
- $att(action_i(c, t_1, \ldots, t_k))$: represents an attempt to perform an action.
- $p_i(t_1, \ldots, t_r)$: generic predicates, used to describe aspects of states of affairs; these predicates are updated as the result of execution of actions.

We differentiate between attempts to perform actions (*att*) and actual actions being performed (*do*): we allow our characters to try whatever they want (via *att* sentences). However, illegal actions may be discarded and/or may cause sanctions, depending on the deontic notions we want or need to implement. The *do* sentences are thus *confirmations* of the *att* sentences. To precisely define computational behaviour we employ the notion of a *state of affairs*:

Definition 17. *A state of affairs $\Delta = \{\varphi_0, \ldots, \varphi_n\}$ is a a finite and possibly empty set of implicitly universally quantified atomic sentences $\varphi_i, 0 \leq i \leq n$, possibly with a deontic modality.*

A state of affairs may contain both atomic sentences and simple deontic sentences. Intuitively, a state of affairs stores the dynamic information on the current situation of the game. A state triggers a collection of *deontic rules*, which in turn update the state Δ generating a new state Δ'.

Whenever an external factor (*e.g.*, an input from a human player) changes the state, rules can be triggered, updating it. States are used to characterise the social commitments of agents, stated in terms of obligations, prohibitions and permissions. We define below the syntax of our deontic rules.

$$
\begin{array}{rcl}
\mathit{Rule} & ::= & \mathit{LHS} \Rightarrow \mathit{RHS} \\
\mathit{LHS} & ::= & \alpha \mid \neg\alpha \mid \mathit{LHS} \wedge \mathit{LHS} \\
\mathit{RHS} & ::= & \mathit{Update} \mid \mathit{RHS} \wedge \mathit{RHS} \\
\mathit{Update} & ::= & \oplus\alpha \mid \ominus\alpha
\end{array}
$$

Fig. 1. Deontic Rules

Definition 18. *A deontic rule, denoted as Rule, is defined by the grammar of Fig. 1, where α is any atomic sentence, possibly with a deontic modality.*

Constructs *Update* are *update operations*: $\oplus\alpha$ inserts α into a state, and $\ominus\alpha$ retracts α from a state.

3.1 Sample Deontic Rules

Sets of deontic rules are not committed to any specific axiom schemata. Rules can even be used to simulate some axiom schemata. For example, axiom (4) in section 2 is simulated using rule $\mathsf{O}(X) \Rightarrow \oplus\mathsf{P}(X)$ (cf. [11,13,12]). Obligations can also generate attempts to perform certain actions. Hence, a sensible general rule to be added to specific theories can be $\mathsf{O}(do(X)) \Rightarrow \oplus att(X)$.

Attempts become effective action executions depending on preconditions. Rules that turn attempts into action executions have the form:

$$
\left(\begin{array}{c} att(action(c, t_1, \ldots, t_m)) \wedge \\ (\bigwedge_i^n \alpha_i) \wedge (\bigwedge_j^u \neg\alpha_j) \end{array}\right) \Rightarrow \left(\begin{array}{c} \oplus do(action(c, t_1, \ldots, t_m)) \wedge \\ \ominus att(action(c, t_1, \ldots, t_m)) \end{array}\right)
$$

In such rules, α_i and α_j are simple deontic sentences. Notice that these sentences can include prohibitions, obligations and permissions. For example, we could have a rule of the form:

$$\left(\begin{array}{r} att(action(c,t_1,\ldots,t_m)) \wedge \\ \mathsf{P}(do(action(c,t_1,\ldots,t_m)) \wedge \\ \neg\mathsf{F}(do(action(c,t_1,\ldots,t_m)) \end{array}\right) \Rightarrow \left(\begin{array}{l} \oplus do(action(c,t_1,\ldots,t_m)) \wedge \\ \ominus att(action(c,t_1,\ldots,t_m)) \end{array}\right)$$

Finally, action execution rules are rules specialised in updating states. An action execution rule has the form $do(action(c,t_1,\ldots,t_m)) \Rightarrow (\bigwedge_i^n \oplus\alpha_i) \wedge (\bigwedge_j^u \ominus\alpha_j)$. Our rule schema is defined as sets of rules precisely written as above.

3.2 Operational Semantics

In this section we provide an operational semantics for our language of deontic rules, which match neatly and perfectly with the conventional operational semantics of forward chaining rules as specified in JSR 94. We show in **Fig. 2** our operational semantics as functions.

1. $\mathbf{s}^*(\Delta, Rules) = \bigcup_{i=0}^{n} \Delta_i' \in \left\{ \mathbf{s}_r'(\Delta, \langle RHS, \Sigma\rangle) \;\middle|\; \begin{array}{l}(LHS \Rightarrow RHS) \in Rules, \\ \mathbf{s}_l^*(\Delta, LHS) = \Sigma, \Sigma \neq \emptyset\end{array} \right\}$
2. $\mathbf{s}_l^*(\Delta, LHS) = \{\sigma \mid \mathbf{s}_l(\Delta, LHS) = \langle\sigma, \mathbf{T}\rangle\}$
3. $\mathbf{s}_l(\Delta, LHS \wedge LHS') = \begin{cases} \langle\sigma, \mathbf{T}\rangle \text{ if } \mathbf{s}_l(\Delta, LHS) = \langle\sigma', \mathbf{T}\rangle \text{ and } \mathbf{s}_l(\Delta, LHS' \cdot \sigma') = \langle\sigma, \mathbf{T}\rangle \\ \langle\emptyset, \mathbf{F}\rangle, \text{ otherwise} \end{cases}$
4. $\mathbf{s}_l(\Delta, \alpha) = \begin{cases} \langle\sigma, \mathbf{T}\rangle \text{ if } \alpha \cdot \sigma \in \Delta \\ \langle\emptyset, \mathbf{F}\rangle \text{ otherwise} \end{cases}$
5. $\mathbf{s}_l(\Delta, \neg\alpha) = \begin{cases} \langle\sigma, \mathbf{T}\rangle \text{ if } \alpha \cdot \sigma \notin \Delta \\ \langle\emptyset, \mathbf{F}\rangle \text{ otherwise} \end{cases}$
6. $\mathbf{s}_r'(\Delta, \langle RHS, \Sigma\rangle) = \bigcup_{i=0}^{m} \Delta_i' \in \{\Delta' \mid \sigma \in \Sigma, \mathbf{s}_r(\Delta, \langle RHS, \sigma\rangle) = \langle\Delta', \mathbf{T}\rangle\}$
7. $\mathbf{s}_r(\Delta, RHS \wedge RHS') = \begin{cases} \langle\Delta'', \mathbf{T}\rangle \text{ if } \mathbf{s}_r(\Delta, \langle RHS, \sigma\rangle) = \langle\Delta', \mathbf{T}\rangle \text{ and} \\ \qquad \mathbf{s}_r(\Delta', \langle RHS', \sigma\rangle) = \langle\Delta'', \mathbf{T}\rangle \\ \langle\emptyset, \mathbf{F}\rangle, \text{ otherwise} \end{cases}$
8. $\mathbf{s}_r(\Delta, \langle\oplus\alpha, \sigma\rangle) = \langle\Delta \cup \{\alpha \cdot \sigma\}, \mathbf{T}\rangle$
9. $\mathbf{s}_r(\Delta, \langle\ominus\alpha, \sigma\rangle) = \langle\Delta \setminus \{\alpha \cdot \sigma\}, \mathbf{T}\rangle$

Fig. 2. Operational Semantics of Deontic Rules

Our operational semantics define an infinite sequence of states of affairs $\langle\Delta_0, \Delta_1, \ldots\rangle$ if $\mathbf{s}^*(\Delta_i, \{\mathsf{R}_1, \ldots, \mathsf{R}_n\}) = \Delta_{i+1}$, that is, Δ_{i+1} (obtained by applying the rules to Δ_i) is used to obtain Δ_{i+2} and so on. **Fig. 3** illustrates how this sequence can accommodate the intervention of players and synthetic characters performing actions. The diagram shows an initial state Δ_0 (possibly empty) that is offered (represented by "$\Rrightarrow$") to a set of human and synthetic characters $\{c_1, \ldots, c_n\}$. They perform actions, adding a record (via "$\updownarrow$") $\{\alpha_1^0, \ldots, \alpha_n^0\}$ of these actions to Δ_0. After the characters add their actions, then the rules are exhaustively applied (represented by "$\overset{*}{\Rightarrow}$") to $\Delta_0 \cup \{\alpha_1^0, \ldots, \alpha_n^0\}$. The resulting state Δ_1 is, on its turn, offered to characters, and so on.

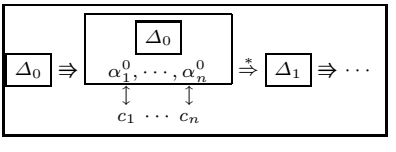


Fig. 3. Semantics as sequence of Δ

Associated to each state of affairs Δ_i we have a model $\mathcal{U}_i = \langle\mathcal{W}, R, V_i\rangle$. For the moment, we leave it to the programmer to ensure any desired relations between subsequent models. We intend in the future to extend this work with more refined rule schemata such that, for instance, if a deontic theory found in state Δ_i is sound and complete with respect to models associated to axiom schemata (1-4)

and (7) then, after the application of a set of deontic rules, the resulting deontic theory found in state Δ_{i+1} is also sound and complete with respect to models associated with the same axiom schemata.

In order to make our discussion more concrete, we present some rules that could be used to implement the behaviour described in section 1. The relevant social norms for the described scene can be captured by the following seven rules:

1. Unless stated otherwise, killing is prohibited: $\top \Rightarrow \oplus \mathsf{F}(do(kill(X,Y)))$
2. A prisoner can be killed: $prisoner(X,Y) \Rightarrow \ominus \mathsf{F}(do(kill(X,Y)))$
3. If X kills Y, then X is identified as the murderer of Y and Y is identified as dead (definition of *kill*): $do(kill(X,Y)) \Rightarrow \ominus do(kill(X,Y)) \wedge \oplus murderer(X,Y) \wedge \oplus dead(Y)$
4. If someone kills X's father, then X has the moral obligation to kill the murderer: $dead(X) \wedge murderer(Y,X) \wedge father(X,Z) \Rightarrow \oplus \mathsf{O}(do(kill(Z,Y)))$
5. (*general deontic rule*) Any obligation generates a permission: $\mathsf{O}(X) \Rightarrow \oplus \mathsf{P}(X)$
6. (*general deontic rule*) Any obligation to execute an action generates an attempt to execute that action: $\mathsf{O}(do(X)) \Rightarrow \oplus att(X)$
7. (*general deontic rule*) Any attempt to execute an action that is permitted and not prohibited leads to the effective execution of that action: $att(X) \wedge \mathsf{P}(do(X)) \wedge \neg \mathsf{F}(do(X)) \Rightarrow \ominus att(X) \wedge \oplus do(X)$

We populate this world with the constants c_w representing the warrior (a good person), c_f representing the warrior's father, and c_m a murderer (a bad person). Using the above rules with $\Delta_0 = \{father(c_f, c_w)\}$, by virtue of rule (1), generates

$$\Delta_1 = \left\{ \begin{array}{lll} father(c_f,c_w), & & \\ \mathsf{F}(do(kill(c_w,c_w))), & \mathsf{F}(do(kill(c_w,c_f))), & \mathsf{F}(do(kill(c_w,c_m))), \\ \mathsf{F}(do(kill(c_f,c_w))), & \mathsf{F}(do(kill(c_f,c_f))), & \mathsf{F}(do(kill(c_f,c_m))), \\ \mathsf{F}(do(kill(c_m,c_w))), & \mathsf{F}(do(kill(c_m,c_f))), & \mathsf{F}(do(kill(c_m,c_m))) \end{array} \right\}$$

If an external action adds to the state the action $do(kill(c_m, c_f))$, this generates $\Delta_2 = \Delta_1 \cup \{do(kill(c_m, c_f))\}$. Triggering the rules with this new state leads to

$$\Delta_3 = \left\{ \begin{array}{lll} father(c_f,c_w), & \mathsf{P}(do(kill(c_w,c_m))), & att(kill(c_w,c_m)), \\ \mathsf{F}(do(kill(c_w,c_w))), & \mathsf{F}(do(kill(c_w,c_f))), & \mathsf{F}(do(kill(c_w,c_m))), \\ \mathsf{F}(do(kill(c_f,c_w))), & \mathsf{F}(do(kill(c_f,c_f))), & \mathsf{F}(do(kill(c_f,c_m))), \\ \mathsf{F}(do(kill(c_m,c_w))), & \mathsf{F}(do(kill(c_m,c_f))), & \mathsf{F}(do(kill(c_m,c_m))), \\ murderer(c_m,c_f), & dead(c_f), & \mathsf{O}(do(kill(c_w,c_m))) \end{array} \right\}$$

Notice that we have in the same state $\mathsf{P}(do(kill(c_w, c_m)))$ and $\mathsf{F}(do(kill(c_w, c_m)))$. Rule (7) makes prohibition override permission in this case, and the *kill*ing does not occur. If we add $prisoner(c_m, c_w)$ to Δ_3, thus generating $\Delta_4 = \Delta_3 \cup \{prisoner(c_m, c_w)\}$, then rule (2) is triggered, and we finally have

$$\Delta_5 = \left\{ \begin{array}{lll} father(c_f,c_w), & \mathsf{P}(do(kill(c_w,c_m))), & att(kill(c_w,c_m)), \\ \mathsf{F}(do(kill(c_w,c_w))), & \mathsf{F}(do(kill(c_w,c_f))), & \\ \mathsf{F}(do(kill(c_f,c_w))), & \mathsf{F}(do(kill(c_f,c_f))), & \mathsf{F}(do(kill(c_f,c_m))), \\ \mathsf{F}(do(kill(c_m,c_w))), & \mathsf{F}(do(kill(c_m,c_f))), & \mathsf{F}(do(kill(c_m,c_m))), \\ murderer(c_m,c_f), & dead(c_f), & \mathsf{O}(do(kill(c_w,c_m))), \\ murderer(c_w,c_m), & dead(c_m) & \end{array} \right\}$$

Clearly, a game would require many more rules to become interesting. For example, we could lift an obligation once the expected results of the corresponding action are observed:

8. $\mathsf{O}(do(kill(X,Y))) \wedge dead(Y) \Rightarrow \ominus\mathsf{O}(do(kill(X,Y)))$

Different game settings can also suggest more sophisticated social norms. For example, a soccer game can be specified, in which prohibited actions can indeed be executed, but are followed by corresponding sanctions (yellow/red card, etc.).

4 Related Work

Apart from classical studies on law, research on norms and agents has been addressed by two different disciplines: sociology and philosophy. On the one hand, socially oriented contributions highlight the importance of norms in agent behaviour (*e.g.*, [14,15]) or analyse the emergence of norms in multi-agent systems (*e.g.*, [16]). On the other hand, logic-oriented contributions focus on the deontic logics required to model normative modalities along with their paradoxes (*e.g.*, [17,18]). The last few years, however, have seen significant work on norms in multi-agent systems, and norm formalisation has emerged as an important research topic in the literature [19,20,21,22,23]. Our approach advances on previous related work by being flexible, yet admitting an efficient implementation.

5 Conclusions, Discussion and Future Work

In this paper we have argued that the efforts of the International Game Developers Association (IGDA) to propose to the game industry interface standards for artificial intelligence techniques using rules must be complemented with automatically verifiable rule schemata that ensure the appropriate implementation of such techniques. We have presented a specific proposal for such schemata, namely, a theory of norms to guide the behaviour of synthetic characters. We have developed a simple, albeit flexible, syntax and proposed an operational semantics for a rule language in which normative aspects (*viz.*, obligations, permissions and prohibitions) are used to specify the behaviour of characters. We have also described a complete rule-based approach to game design.

Our proposal encourages high-level design principles based on metaphors and paradigms humans relate with, in this case, normative concepts such as obligations, permissions and prohibitions. Rather than having game designers pursuing their ideas in the abstract, possibly using simple storyboards and graphic props, and then transmitting their ideas to implementors, rules can "short-circuit" this two-step process. If designers make use of rules (via suitable support tools) and since rules capture behaviours, then the rules are an initial prototype which can then be used to guide the final, more efficient, implementation. Even if rules were to be "discarded" after exploration, they can still serve as a means to communicate the design decisions to implementors.

We want to investigate (semi-)automatic means to check properties of deontic rules. In particular, we want to explore the formal connections between the model-theoretic and operational semantics sketched in Section 3.2, and propose algorithms to check if a set of deontic rules is "feasible", that is, they admit

at least one model. We also want to study other properties and how these can be checked; for instance, we want to provide guarantees that all obligations are fulfilled and that prohibitions are never infringed. Such checks could be done "a posteriori", after the set of deontic rules have been finalised, or *during* their preparation, as editing and checking operations can be *interleaved*.

We envisage a framework for game design centred around rule schemata. Game designers create their rules using a suite of tools for editing and checking them. Since more than one designer can be involved in the rule preparation, the editing and management of rules must allow for collaborative design. Ours is a distributed framework using a shared blackboard implemented via a tuple space. Once rules are created and made available, they can be "tried out" immediately: synthetic characters put the rules to use, as the operational semantics of rules defines precise behaviours. The states of affairs representing the operational semantics can also be stored on the blackboard, thus providing explicit and comprehensive descriptions of how the game evolved.

Such framework encourages design experimentation as it short-circuits the design-followed-by-implementation process. Our proposal should encourage high-level design principles using a metaphor (*viz.*, deontic notions) humans relate to, rather than low-level implementational details scattered over code of a programming language. We highlight the fact that rules can be used by various people, from artists to programmers, and also by software artifacts.

References

1. Laird, J.E., van Lent, M.: Human-level AI's Killer Application: Interactive Computer Games. In: Procs. 17th Nat'l Conf. on A.I. (AAAI'2000), Austin, Texas,U.S.A., AAAI Press (2000)
2. Combs, N., Ardoint, J.L.: Declarative vs. Imperative Paradigms in Games AI (2005) Available at `http://www.roaringshrimp.com/WS04-04NCombs.pdf`, Last revised in June 2005.
3. Johnson, D., Wiles, J.: Computer Games with Intelligence. In: Procs. 10th IEEE Int'l Conf. on Fuzzy Systems, Melbourne, Australia, IEEE (2001)
4. Sweetser, P.: Current AI in Games: a Review (2002) Unpublished manuscript, available at `http://www.itee.uq.edu.au/~penny/_papers/Game%20AI%20Review.pdf`, University of Queensland, Last accessed 05 March 2006, 16:15GMT.
5. International Game Developers Association: Report of the IGDA's Artificial Intelligence Interface Standards Committee (2005) Available at `http://www.igda.org/ai/ report-2005/report-2005.html`, Last accessed 05 March 2006, 16:03GMT.
6. Java: Community Development of Java Technology Specifications: Java Specification Requests (2005) Available at `http://www.jcp.org/en/jsr/all`, Last accessed 05 March 2006, 15:28GMT.
7. Valente, A.: Legal Knowledge Engineering: A Modelling Approach. IOS Press, Amsterdam, The Netherlands (1995)
8. Mally, E.: Grundgesetze des Sollens: Elemente der Logik des Willens. Leuschner & Lubensky, Graz, Austria (1926)
9. von Wright, G.: Deontic Logic. Mind **60** (1951) 1–15

10. Aqvist, L.: Deontic Logic. In: Handbook of Philosophical Logic. Volume II: Extensions of Classical Logic. Kluwer, Dordrecht (1984)
11. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.: A Distributed Architecture for Norm-Aware Agent Societies. In: Procs. Int'l Workshop on Declarative Agent Languages & Technologies (DALT 2005), New York, USA. Volume 3904 of LNAI. Springer-Verlag, Berlin (2006)
12. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.: Norm Oriented Programming of Electronic Institutions. In: Procs. 5th Int'l Joint Conf. on Autonomous Agents & Multiagent Systems (AAMAS'06), Hakodate, Japan, ACM Press (2006)
13. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.: A Rule-based Approach to Norm-Oriented Programming of Electronic Institutions. ACM SIGecom Exchanges **5** (2006) 33–40
14. Conte, R., Castelfranchi, C.: Understanding the Functions of Norms in Social Groups through Simulation. In: Artificial Societies. The Computer Simulation of Social Life, UCL Press (1995)
15. Tuomela, R., Bonnevier-Tuomela, M.: Norms and Agreement. European Journal of Law, Philosophy and Computer Science **5** (1995) 41–46
16. Shoham, Y., Tennenholtz, M.: On Social Laws for Artificial Agent Societies: Off-line Design. Artificial Intelligence **73** (1995) 231–252
17. López y López, F., Luck, M.: A Model of Normative Multi-Agent Systems and Dynamic Relationships. Volume 2934 of LNAI., Springer-Verlag (2004)
18. von Wright, G.: Norm and Action: A Logical Inquiry. Routledge and Kegan Paul, London (1963)
19. Artikis, A., Kamara, L., Pitt, J., Sergot, M.: A Protocol for Resource Sharing in Norm-Governed Ad Hoc Networks. Volume 3476 of LNCS. Springer-Verlag (2005)
20. Boella, G., van der Torre, L.: Permission and Obligations in Hierarchical Normative Systems. In: Procs. ICAIL 2003, ACM Press (2003)
21. Dignum, F.: Autonomous Agents with Norms. A. I. & Law **7** (1999) 69–79
22. Michael, L., Parkes, D.C., Pfeffer, A.: Specifying and monitoring market mechanisms using rights and obligations. In: Proc. AMEC VI. (2004)
23. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Implementing Norms in Multiagent Systems. Volume 3187 of LNAI., Springer-Verlag (2004)

Selecting a Feature Set to Summarize Texts in Brazilian Portuguese

Daniel Saraiva Leite and Lucia Helena Machado Rino

Departamento de Computação, UFSCar
CP 676, 13565-905 São Carlos, SP, Brazil
Núcleo Interinstitucional de Lingüística Computacional
http://www.nilc.icmc.usp.br
{daniel_leite; lucia}@dc.ufscar.br

Abstract. This paper presents a novel approach to combining features for training an automatic extractive summarizer of texts written in Brazilian Portuguese. The approach aims at both diminishing the effort of classifying features that are representative for Automatic Summarization and providing more informativeness for the summarizer to decide which text spans to include in an extract. Finding a balanced set of features is explored through WEKA. We discuss several ways of modifying the feature set and show how automatic feature selection may be useful for customizing the summarizer.

1 Introduction

Extractive methods for Automatic Summarization (AS) depend mostly on empirical or statistical techniques that aim at recognizing which text spans are relevant to include in the final extract. Oppositely, unimportant text spans can also be filtered from the source text. Several features are acknowledged to play an important role in such a process, usually addressing different types of information, such as grammatical or position features, words in titles, or even functional information conveyed by the source text. Mani [15] and others (e.g., [4]; [11]) suggest a general algebraic formula pinpointing the complex combination of those features, resulting in a general reasoning model. Roughly, the formula adds up all the considered features, which are usually weighted according to their influence in the context, or to the summarizing model itself. Defining weights to adequately rank features is the main bottleneck in such a ranking strategy. The reasons for this difficulty are the following: features may be genre- or domain-dependent, and the more features considered, the more complex their combination is. Actually, the complexity is due to the absence of guidelines to put together a set of features that, isolated, are agreeably relevant to AS. So, finding a nice balance between features is very subtle, making knowledge engineering more complex as more features are considered.

In this paper, we focus on extracting excerpts from texts written in Brazilian Portuguese (BP) especially addressing SuPor ([17]; [20]) an environment for summarization of texts in Portuguese that must be hand-customized through the selection of up to seven features in order to build extracts. One of the SuPor summarizing strategies

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 462 – 471, 2006.

outperformed six other summarizers when assessed on a test corpus of news texts [21]. However, the reported performance did not allow determining which of its features played a significant role in representing or adequately informing the content of the source texts. Reasons were clearly pinpointed in [20]: its f-measure was still under average and it performed too close to ClassSumm [13], one of the systems that considered sixteen features, instead of the five ones used in SuPor. Most importantly, SuPor usability is quite troublesome, due to the effort imposed to its user for selecting a satisfactory feature set.

Pursuing the goal of finding a better way of selecting features, and thus, aiming at determining a good feature set, we explore how to take advantage of WEKA [23] facilities (http://www.cs.waikato.ac.nz/~ml/weka/ [May 29]). WEKA is an open source environment for machine learning which embeds several techniques for data pre-processing, classification, regression, clustering, association rules, and visualization.

In Section 2 we outline the main features of SuPor and its underlying AS methodology, pinpointing some of its drawbacks. In Section 3 we present the new approach to selecting features, followed by its assessment (Section 4). Final remarks are presented in Section 5.

2 The SuPor Environment

SuPor combines distinct AS models originally applied to English texts, using a machine learning approach [11] that employs a Naïve-Bayes classifier to rank sentences according to their likelihood of being in an extract. Thus, the greater the likelihood, the more relevant the sentence is to be included in the extract. Apart from the language-dependent resources (stoplist and lexicon) and tools (stemmer and tagger), SuPor incorporates linguistic and non-linguistic features that, combined, delineate a method to summarize texts in BP. Possible features and their combination for training and extraction are described here.

2.1 SuPor Features

Features in SuPor can be chosen amongst three full, fixed feature subsets that depict complete AS methods or other four corpora-based features, as follows:

The Lexical Chains Method [1]. This computes the connectedness between words based upon the existence of lexical chains [8] in the source text. A lexical chain is a set of words related through cohesive means (e.g., synonymy/antonym or hiperonimy/hiponimy). The idea of this method is that the stronger the lexical chain, the more it conveys important concepts of the source text.

After computing lexical chains, three sets of sentences are produced through the following heuristics: (1) selecting every sentence S of the source text based on each member M of every strong lexical chain of the set formerly computed. S is the sentence that contains the first occurrence of M; (2) applying the former one only to representative members of a strong lexical chain. A representative member of a lexical chain is that whose frequency is greater than the average of all words in the chain; (3) selecting representative lexical chains of every topic of the source text.

If a sentence of the source text appears in at least one of those sets, the ‘Lexical Chains’ feature will be ‘True’ for the sentence; ‘False’, otherwise.

The Text Relationship Map Method [22]. This performs similarly to the previous one, but considering paragraphs instead of sentences. It builds the so-called relationship map of a source text as a graph that represents its degree of cohesion. Graph nodes are paragraphs themselves and edges signal their connectiveness in the text. This is expressed by the similarity scores of the paragraphs. Summarizing may be carried out in three different ways, depending on the traversal mode of the graph. The first, dense, path is the one that selects the bushy nodes of the graph, i.e., only those paragraphs that are significantly dense, even if they are not semantically inter-related. The second, deep, path focuses on the opposite, i.e., only on paragraphs that are semantically inter-related, aiming at a more coherent extract. However, it does not convey a broad range of topics. Finally, the third path (segmented) aims at overcoming the bottlenecks of the others, by addressing distinct topics of the source text and, thus, both coherence and coverage.

Similarly to the 'Lexical Chains' feature, if a sentence of the source text appears in a paragraph selected by at least one of such paths, 'Text Relationship Map' will be 'True' for that sentence; 'False' otherwise.

The Importance of Topics Method [12]. This method also aims at identifying the main topics of the source text, which will then drive sentence extraction. This is accomplished by selecting, for each topic, a number of sentences proportional to the importance of that topic. Additionally, to determine which ones, from all the sentences related to the topic will be finally extracted, sentence importance is calculated through its similarity to the topic centroid [12].

According to a particular cutoff number of sentences of each topic, the 'Importance of Topics' feature will be 'True' if the sentence is selected; 'False' otherwise.

Sentence Length. If sentence length is above the cutoff of five words, its 'Sentence Length' feature will be 'True'. Otherwise, it will be 'False'.

Sentence Location. If a sentence figures in a sentence set of the first 10% and the last 5% sentences of the source text, its corresponding feature will be 'True'; it will be 'False' otherwise.

Proper Nouns. If a sentence contains a number of proper nouns above a threshold, the 'Proper Nouns' feature will be 'True'. Otherwise, it will be 'False'.

Words Frequency. For each sentence, a score defined by the sum of its words frequency is computed. If a sentence score is greater than a specific threshold, its feature will be 'True'. Otherwise, it will be 'False'.

Actually, some of those methods split up into two features, because the user can choose between distinct preprocessing methods. For 'Lexical Chains', preprocessing regards text segmentation into topics, which is achieved either through tiling the text [7] or delimiting paragraphs as topic units themselves. For the features 'Text Relationship Map', 'Importance of Topics' and 'Words Frequency', the user can also choose between preprocessing the source texts, now through stemming or 4-gramming. Due to such preprocessing variations, to customize a summarizer the user has actually available

eleven distinct features. So, potentially any desirable feature subset can be arbitrarily chosen. This is quite problematic to manage, even by an expert user, as pinpointed in [21].

2.2 Training and Extraction in SuPor

SuPor training corpus comprises both, source texts and corresponding ideal extracts. Every sentence from the source texts is represented in the training dataset as a tuple of binary features chosen by the user. Such tuples are labeled with respect to the class ('Present' or 'Not Present in Summary') as follows: if the corresponding sentence matches a sentence of its ideal extract, then the class label is valued 'True' (i.e., Present in Summary); it is valued 'False' otherwise.

The training dataset is used to compute the probabilities used by the Bayesian classifier to rank sentences that are more likely to be selected to compose the extract. In other words, the probability for the 'True' class is computed. Top-ranked sentences are then selected, according to the user-defined compression rate.

2.3 Remarks on SuPor Environment Issues

Keeping unchanged the extraction and training models and considering the vulture of combining any features out of the eleven available ones, SuPor customization is quite complex to address summarizing options. Simply considering the feature 'Words Frequency', for example, if there were two sentences scoring above the threshold, even if one of them ranked higher than the other, this would not be taken into account. Actually, both would be just 'True'.

Ideally, features conveying such differences between sentences could allow for a better performance of the system. This is the focus of our new approach in finding a simpler way of training and bringing more informativeness to SuPor. In differentiating in a more fine-grained way the involved features, we aim at better analyzing the real contribution of the features to the final results, as discussed next.

3 A New Approach to Selecting a Feature Set

The goal of the approach reported in this article is to overcome both SuPor most serious drawbacks: the use of binary values to define a proper feature set for summarization and the struggle imposed to the user in customizing it. We propose a new summarizer, hereafter named SuPor-v2, which addresses more informative features and the means to perform automatic feature selection. This new approach aims at improving the system performance and scalability.

3.1 Boosting Features Informativeness

To improve the expressiveness of a feature set, we assumed that features values would not be binary anymore. We used categorical and numeric values instead. Categorical domains are defined for 'Lexical Chaining', 'Text Relationship Map', and 'Sentence Location', as show tables 1 to 3. The new categories provide a fine-grained setting derived from the original methods, for the first two features, or from the functionality of

the feature, for 'Sentence location'. So, Table 1 shows that eight distinct symbolic labels, mirroring diverse heuristics, apply to 'Lexical Chaining', whilst eight symbolic labels signal possible paths to identify paragraphs to compose an extract (Table 2). 'Sentence Location' was refined according to the most probable positions of relevant information suggested by others (e.g., [4], [11]).

Table 1. Possible labels for 'Lexical Chains' features

Label	Description
False	No heuristics have recommended the sentence
H1	Only first heuristic has recommended the sentence
H2	Only second heuristic has recommended the sentence
H3	Only third heuristic has recommended the sentence
H1+H2	Both first and second heuristics have recommended the sentence
H1+H3	Both first and third heuristics have recommended the sentence
H2+H3	Both second and third heuristics have recommended the sentence
H1+H2+H3	All heuristics have recommended the sentence

Table 2. Possible labels for 'Text Relationship Map' features

Label	Description
False	No paths include the sentence
C1	Only first path includes the sentence
C2	Only second path includes the sentence
C3	Only third path includes the sentence
C1+C2	Both first and second paths include the sentence
C1+C3	Both first and third paths include the sentence
C2+C3	Both second and third paths include the sentence
C1+C2+C3	All paths include the sentence

Table 3. Possible labels for 'Sentence Location' feature

Label	Paragraph Position within the text	Sentence position within the paragraph
II	Initial	Initial
IM	Initial	Medial
IF	Initial	Final
MI	Medial	Initial
MM	Medial	Medial
MF	Medial	Final
FI	Final	Initial
FM	Final	Medial
FF	Final	Final

Apart from such complex features, the others were related only to more fine-grained numeric domains. 'Importance of Topics', for example, signals the relevance of the sentences of a source text using two measures (as reported in Section 2.1), both defined in the range [0..1]. In our approach, we defined relevance by calculating the harmonic mean of those measures. This preserves the notion that both measures are

important. At the same time, sentences related to unimportant topics or sentences (i.e., those with low similarity with the centroids of their related topics), are penalized. The remaining features associated with numeric values were refined as follows:

- 'Sentence Length' is valued by the normalized length of a sentence, with respect to the longest sentence in the text.
- 'Proper Nouns' values signal the normalized count of proper nouns in a sentence.
- 'Words Frequency' mirrors the score computed for each sentence, normalized with respect to the top-scored sentence.

3.2 Automatic Feature Selection

We address automatic feature selection (FS) through a filter approach, which aims at selecting relevant features by analyzing only the training set [23]. In machine learning, filtering through FS is a previous process to machine learning itself. The learnt feature set will configure our so-called SuPor-v2.

We perform correlation feature selection, or CFS, due to its following advantages [6]: (1) It is quite fast; (2) The user is not required to specify a limited number of features; (3) It significantly improves accuracy of classifiers in most cases. Like other FS filtering approaches, CFS is strongly based upon the entropy measure. Overall, the method looks for feature subsets highly correlated with the class, at the same time that they convey few redundancies within the set itself. Through such a method, a feature set was defined. Its assessment is reported in the next section.

4 Assessing the Feature Set for Brazilian Portuguese Source Texts

We carried out two experiments, both blackbox-based, in that only the systems outputs were compared. Only the informativeness of the extracts was assessed, as reported below. Firstly, SuPor-v2 was evaluated aiming at analyzing the influence of CFS in selecting features. In this experiment another classifier was also employed, which was compared to the Naïve-Bayes classifier.

The second experiment consisted of replicating SuPor assessment on seven systems [21] including, now, the best strategy found in the first experiment. The aim here was to verify if indeed SuPor-v2 improved the informativeness of the generated extracts. Important to notice is that, in adopting the same assessment setting, we did not compute over the results of the other systems.

As formerly, TeMário [18] was used as the test corpus. It actually amounts to three different text sets: 100 rough news source texts, their corresponding manual summaries, and extracts automatically produced. Both manual summaries and extracts are considered our ideal, reference corpora, for comparison with SuPor-v2 results.

4.1 Assessing SuPor-v2 Strategies

Aiming at verifying whether the Naïve-Bayes model suited better our needs, we also considered the use of the C4.5 classifier [19], which employs a decision tree for classification. Both Naïve-Bayes and C4.5, though, do not handle numeric features, which are

present in our new feature set. To overcome that, we employed two different techniques: *Supervisioned Discretization* [5] and *Kernel Density Estimation* [10], hereafter named KDE. The first maps numeric feature values onto discrete, categorical, ones. So, no numeric features are handled any longer. The second technique is only suitable for Naïve-Bayes. It allows approximating the probabilities required by the classifier through a sum of the so-called kernels, which are Gaussians centered around data points.

The above methods are readily implemented (thus, available) in WEKA. Roughly, SuPor-v2 was built coupling together SuPor and WEKA. Again, the only metrics for assessing SuPor-v2 were precision (P=number of relevant sentences/length of the extract) and recall (R=number of relevant sentences/length of the ideal extract). Extracts were produced under a 30% compression rate. The f-measure was thus derived from the others (F=2*P*R/(P+R)). For every possible combination of such techniques, a 10-fold cross validation evaluation (each fold comprising ten texts) was performed. Table 4 shows precision, recall and f-measure averages for both classifiers ('No' signals that processing is based upon all the eleven features).

Table 4. Assessing distinct machine learning strategies

Model	Classifier	Numeric handling	Feature Selection	Recall (%)	Precision (%)	F-measure (%)
M1	Naïve-Bayes	KDE	No	43,9	47,4	45,6
M2			CFS	42,8	46,6	44,6
M3		Discretization	No	42,2	45,8	43,8
M4			CFS	42,0	45,9	43,9
M5	C4.5	Discretization	No	37,7	40,6	39,1
M6			CFS	40,2	43,8	41,9

Overall, the best classifier for SuPor-v2 was the Naïve-Bayes one, as formerly. This is not surprising, when considering its optimal performance for probability-based ranking [25]. Although Naïve-Bayes probabilities are not statistically precise [3], for our purposes it is significant, since we focus only on ranking. Moreover, it outperforms C4.5, which actually attempts to produce probability estimates by computing the relative frequency of each class in a leaf. Thus, all new instances accordingly classified (i.e., in the same leaf) will have an equal probability. This may be problematic, especially when AS takes place. If the generated C4.5 tree is small, most frequently there will be few leaves. Therefore, many sentences will present the same probability. This problem impairs substantially the capacity of C4.5 in representing accurate rankings. Naïve-Bayes, on the contrary, can represent a wider range of distinct class probabilities [25].

Concerning numeric handling, KDE allows a better performance than the discretization method when Naïve-Bayes is considered. Regarding CFS, we did not obtain the expected results. For Naïve-Bayes, the results are slightly better when all features are used ('No' option in Table 4). C4.5 seems to perform slightly worse without CFS. Such results may suggest that a stable feature set was achieved. So, no feature selection would be needed anymore, to customize SuPor-v2

4.2 Comparing SuPor-v2 with the Other Summarizers

In comparing SuPor-v2 with the other systems, only our best model was considered, i.e., M1 in Table 4. Since the setting for assessing SuPor-v2 was identical to the one defined for the other systems, we simply included the new values into the very same table found in [21], resulting in Table 5.

Table 5. Systems performance (in %)

Systems	Precision (%)	Recall (%)	F-measure (%)	% over random
SuPor-v2	47.4	43.9	45.6	47
SuPor	44.9	40.8	42.8	38
ClassSumm	45.6	39.7	42.4	37
From-top	42.9	32.6	37.0	19
TF-ISF-Summ	39.6	34.3	36.8	19
GistSumm	49.9	25.6	33.8	9
NeuralSumm	36.0	29.5	32.4	5
Random order	34.0	28.5	31.0	0

Although there is a difference in f-measure of only c.a. 3-point percentual between SuPor and SuPor-v2, the performance of this system is still significant, considering that increasing such percentiles is quite difficult. Also, it suggests that non-binary features may help selecting sentences more precisely, to include in an extract. SuPor-v2 also reassures the utility of using Naïve-Bayes for classification. Actually, Naïve-Bayes is the coinciding method of the topmost three systems in Table 5.

5 Final Remarks

Clearly, considering more informative features and, thus, bringing more information into the machine learning model indicates a way of improving extractive AS. We argue that even the shallow features can be powerful, provided that they are more expressively represented, i.e., with a range of features different from the binary one.

It is also worthy considering that the machine learning approach to AS is quite different from most common classification tasks, because, instead of considering just the predicted class, a rank is desirable. So, empowering a classifier that does not focus on ranking may not be the utmost goal to pursue. Using Naïve-Bayes may well signal the potentiality of probability-based classifiers for AS. Building on this, even Bayesian Networks (e.g., [2]) can be adopted, providing a good testbed for overcoming some of the drawbacks of the Naïve-Bayes classifier [23]. In the future, other methods for numeric handling shall also be explored. Varying discretization methods may also be a good perspective, since they seem to apply better to Naïve-Bayes classifiers (e.g., [24]; [9]).

Finally, although the reported work has been applied to Brazilian Portuguese, the approach to selecting and enriching a feature set to provide more informativeness for a summarizer is not language-dependent.

Acknowledgements

This work has been funded by CNPq.

References

1. Barzilay, R., Elhadad, M.: Using Lexical Chains for Text Summarization. In: Mani, I., Maybury, M. T. (eds.): Advances in Automatic Text Summarization. MIT Press (1997) 111-121
2. Cooper, G., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9 (1992) 309-347
3. Domingos, P., Pazzani, M.: Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In: L. Saitta (ed.): Proc. of the 13th International Conference on Machine Learning. Morgan Kaufmann, Bari, Italy (1996) 105-112
4. Edmundson, H.P.: New methods in automatic extracting. Journal of the Association for Computing Machinery 16 (1969) 264-285
5. Fayyad, U., Irani, K.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: Bajcsy, R. (ed.): Proc. of 13th International Joint Conference on Artificial Intelligence. Morgan Kaufmann, Chambery, France (1993) 1022-1027
6. Hall, M. A.: Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In: Langley, P. (ed.): Proc. of 17th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA (2000) 359-366
7. Hearst, M.A.: TextTiling: A Quantitative Approach to Discourse Segmentation. Technical Report 93/24. University of California, Berkeley (1993)
8. Hoey, M.: Patterns of Lexis in Text. Oxford University Press (1991)
9. Hsu, C. N., Huang, H. J., Wong, T. T. Why discretization works for naive Bayesian classifiers. In: Langley, P. (ed.): Proc. of 17th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA (2000) 309-406
10. John, G., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Besnard, P., Hanks, S. (eds.): Proc. of the 11th Conference on Uncertainty in Artificial Intelligence. Quebec, Canada (1995) 338-345
11. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Fox, E. A., Ingwersen , P., Fidel, R. (eds.): Proc. of the 18th ACM-SIGIR Conference on Research & Development in Information Retrieval. Seatlle, WA (1995) 68-73
12. Larocca Neto, J., Santos, A.D., Kaestner, A.A., Freitas, A.A.: Generating Text Summaries through the Relative Importance of Topics. In: Monard, M. C., Sichman, J. S. (eds.): Proc. of 15th Brazilian Symposium on Artificial Intelligence (SBIA'00). Lecture Notes in Artificial Intelligence, No. 1952, Springer-Verlag (2000) 300-309
13. Larocca Neto, J., Freitas, A.A., Kaestner, C.A.A.: Automatic text summarization using a machine learning approach. In: Bittencourt, G., Ramalho, G. L. (eds.): Proc. of 16th Brazilian Symposium on Artificial Intelligence (SBIA'02). Lecture Notes in Artificial Intelligence, No. 2057, Springer-Verlag (2002) 205-215
14. Luhn, H.: The automatic creation of literature abstracts. IBM Journal of Research and Development 2 (1958) 159-165
15. Mani, I.: Automatic Summarization. John Benjamin's Publishing Company (2001)
16. Mani, I., Maybury, M.T.: Advances in Automatic Text Summarization. MIT Press (1999)

17. Módolo, M.: SuPor: an Environment for Exploration of Extractive Methods for Automatic Text Summarization for Portuguese (in Portuguese). MSc. Dissertation. Departamento de Computação, UFSCar (2003)
18. Pardo, T.A.S., Rino, L.H.M.: TeMário: A corpus for automatic text summarization (in Portuguese). NILC Tech. Report NILC-TR-03-09 (2003)
19. Quinlan, J.R: C4.5 Programs for machine learning. Morgan-Kaufman, San Mateo (1993)
20. Rino, L.H.M., Módolo, M.: SuPor: An environment for AS of texts in Brazilian Portuguese. In: Vicedo, J. L., Martínez-Barco, P., Muñoz, R. et al. (eds.): Advances in Natural Language Processing. Lecture Notes in Computer Science, No. 3230, Springer-Verlag, Heidelberg (2004) 419-430
21. Rino, L.H.M., Pardo, T.A.S, Silla Jr., C.N., Kaestner, C.A.A., Pombo, M.: A Comparison of Automatic Summarizers of Texts in Brazilian Portuguese. In Bazzan, A.L.C., Labidi, S. (eds.): Proc. of 17th Brazilian Symposium on Artificial Intelligence (SBIA'04). Lecture Notes in Artificial Intelligence, No. 3171, Springer-Verlag (2004) 235-244
22. Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic Text Structuring and Summarization. Information Processing & Management 33 (1997) 193-207
23. Witten, I. H, Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd ed. Morgan Kaufmann, San Francisco (2005)
24. Yang, Y.: Discretization for Naive-Bayes Learning. PhD Thesis, Monash University (2003)
25. Zhang, H., Su, J.: Naive Bayesian Classifiers for Ranking. In: Boulicaut, J., Esposito, F., Giannotti, F., Pedreschi, D. (eds.): Proc. of the 15th European Conference on Machine Learning (ECML'04). Lecture Notes in Computer Science, No. 3201, Springer, Pisa, Italy (2004) 501-512

Word Sense Disambiguation Based on Word Sense Clustering

Henry Anaya-Sánchez[1], Aurora Pons-Porrata[1], and Rafael Berlanga-Llavori[2]

[1] Center of Pattern Recognition and Data Mining, Universidad de Oriente, Cuba
henry@csd.uo.edu.cu, aurora@app.uo.edu.cu
[2] Universitat Jaume I, Castellón, Spain
berlanga@lsi.uji.es

Abstract. In this paper we address the problem of Word Sense Disambiguation by introducing a knowledge-driven framework for the disambiguation of nouns. The proposal is based on the clustering of noun sense representations and it serves as a general model that includes some existing disambiguation methods. A first prototype algorithm for the framework, relying on both topic signatures built from WordNet and the Extended Star clustering algorithm, is also presented. This algorithm yields encouraging experimental results for the SemCor corpus, showing improvements in recall over other knowledge-driven methods.

1 Introduction

Word Sense Disambiguation (WSD) is the general task of deciding the appropriate sense for a particular use of a polysemous word given its textual context. Despite of the intermediate nature of this task, it defines an essential research area in Natural Language Processing that contributes to almost all intelligent text processing applications (e.g. Machine Translation, Information Extraction, Question Answering, Text Summarization, etc.).

The task of WSD can be specialized according to the sense definitions. For instance, word sense induction refers to the process of discovering different senses of an ambiguous word without prior information about the inventory of senses [20]. On the other hand, there are two major approaches for the disambiguation when predetermined sense definitions are provided: data-driven (or corpus-based) and knowledge-driven WSD. Data-driven methods are supervised due to the fact that they learn over hand-tagged samples. Unlike them, knowledge-driven methods use a background knowledge source, avoiding the use of samples. Currently, lexical resources like WordNet [12] constitute the referred source in most cases.

Both, sense induction and data-driven approaches suffer from the sensitivity of the data together with the adquisition and real-world data problems (e.g. completeness, correctness, etc.) [7], which give a domain-specific character to these approaches. These circumstances along with the availability of widely tested lexical resources make preferable the use of knowledge-driven methods for WSD, even when supervised approaches clearly outperfom them on well-known test sets (e.g. SENSEVAL corpus).

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 472–481, 2006.

Most of knowledge-driven methods have a similar behaviour: they try to match a textual context against the knowledge source, then select the best match and finally retrieve from it the suitable senses for the context constituents. Therefore, the main differences among these methods stem from the knowledge source, the relations used to perform the match and the best match selection procedure.

For example, the Lesk method [8] is based on counting word overlaps between dictionary definitions and the context of an ambiguous word. In [4] simulated annealing is used to handle the combinatorial explosion of the Lesk method. Recently, several approaches such as [2], [5], [15] and [17] consider lexical relations (like hypernymy/hyponymy) among context elements[1]. Other works rely on the Web as knowledge source and use syntactic or text-proximity relations [18].

In this paper, we address the problem of word sense disambiguation by introducing a knowledge-driven framework with a first prototype to disambiguate nouns. Our approach is based on the clustering of sense representations as a natural way to capture the reflected cohesion among the words of a textual unit. Starting from an initial cluster distribution of all possible senses, the algorithm selects groups of senses and discards others by matching the textual context against the clusters. The selected senses are grouped again and the process is repeated until a certain disambiguation criterion holds. Finally, words are disambiguated with the remaining senses.

To the best of our knowledge, clustering algorithms have been explicitly used in the WSD area for two main purposes. The primary one consists of clustering textual contexts to represent different senses in corpus-driven WSD (e.g. [14]) and to induce word senses (e.g. [16], [3]). Other works (like [10] and [1]) cluster fine-grained word senses into coarse-grained ones. Hence, this paper shows a novel way of using clustering in this field.

In addition, our proposal aims to serve as a general model that includes some existing knowledge-driven methods and, at the same time, attempts to be a new method with better performance. Despite the fact that we only treat in this paper the disambiguation of nouns, the approach can be extended to consider any kind of words right away.

The rest of the paper is organized as follows. First, Section 2 specifies the proposed framework. In Section 3 a first prototype is presented. Section 4 describes some experiments (carried out on SemCor corpus) and its results. Finally, Section 5 is devoted to offer some considerations and future work as conclusions.

2 A Knowledge-Driven Framework for WSD

Usually, text processing applications require the disambiguation of a specific subset of words (e.g. the most frequent nouns), instead of an exhaustive full-text word disambiguation. Following this idea, the goal of our framework is the disambiguation of a finite set of nouns N given a textual context T.[2]

[1] These relations are present in well-known lexical databases like WordNet.

[2] Here, we do not restrict the elements of N to be in T.

Our framework comprises the following elements:

i. a representation for senses, which is provided by the knowledge source,
ii. a clustering algorithm capable to group related sense representations,
iii. a matching function for comparing a sense cluster with the textual context,
iv. a filtering function for selecting sense clusters relying upon the previous function and
v. a stopping criterion for ensuring the termination of the disambiguation process.

The idea behind this is that the noun senses in the given context must be related by means of a certain – and possibly complex – relation. As we are not interested in the precise relation definition but in the senses it links, we suggest the use of a clustering algorithm. The role of this algorithm consists of putting together the related senses into cohesive clusters. Assuming that each sense cluster represents a possible meaning for the set of its constituent nouns, the rigth ones must be identified using the textual context. Thus, it is necessary a filtering method to select those clusters that match the best with the context. Due to the intrinsic difficulties in the modelling of the mentioned relation we propose an iterative process to refine the clustering. Hence, it is also required a stopping criterion. The general steps of the framework are shown in Algorithm 1.

Algorithm 1. Framework for the disambiguation of the set of nouns N in the textual context T

Input: The finite set of nouns N and the textual context T.
Output: The disambiguated noun senses.
Let S be the set of all senses of nouns in N;
repeat
$G = group(S)$
$G' = filter(G, T, matching\text{-}function)$
$S = \cup_{g \in G'} \{s | s \in g\}$
until *stopping-criterion*
return S

In this algorithm, the functions *group*, *matching-function*, *filter* and *stopping-criterion* correspond to the framework components ii., iii., iv. and v., respectively. By specifying these functions and the sense representation, different disambiguation algorithms can be obtained from this framework. Moreover, some existing knowledge-driven disambiguation methods can be seen as instances of this proposal. For example, the Specification Marks method [13] can be expressed by representing each sense with the set of all its hypernym synsets, and by defining the other framework components as follows. The clustering algorithm builds the set of clusters $\{subsume(c) \cap S | c \in subsumed\_by\_some\_in(S)\} \cup \{\{s\} | s \in S\}$, where $subsumed\_by\_some\_in(S)$ is the set of all WordNet senses that are subsumed by senses in S and $subsume(c)$ is the set of senses subsuming c, according to the hyponym relation. The matching function associates to each cluster the

number of context words having at least one sense in it. Note that here the textual context T coincides with N. The component *filtering-function* firstly selects for each noun n in N the clusters having the greatest matching score that contains just one sense of n. If there is only one selected cluster for n, the output of the function will include the singleton containing this sense. Otherwise, the output of the function will include the singletons (containing senses of n) obtained by applying five heuristics over the selected clusters. Finally, in this method only one framework iteration is required, therefore *stopping-criterion* is the constant *True*.

Similarly, the Conceptual Density algorithm [2] can be derived from the framework by using the WordNet hypernym relation to provide sense representation. Also, $group(S)$ must be defined as the trivial clustering $\{\{s\}|s \in S\}$, the matching function as the function that assigns a score to each cluster based on the conceptual density formula, the filtering function as the function that selects the cluster having the greatest matching score for each noun in N, and the stopping criterion as the constant *True*.

3 A New WSD Method

In this section we introduce our first prototype algorithm by defining its framework components.

Sense representation: The algorithm uses Topic Signatures [9] as representations for the WordNet nominal senses. The topic signature of a noun sense s is a finite set $\{< t_1, w_1 >, ..., < t_m, w_m >\}$, where each t_i is a term (unigram, bigram or trigram) highly correlated to s with association weight w_i. These signatures are represented using the Vector Space Model (VSM) [19]. Also, a correspondence between a sense and the noun in N it represents is assumed.
Clustering algorithm: We adopt the Extended Star Clustering Algorithm [6], which builds overlapped and star-shaped clusters. Each cluster consists of a star and its satellites, where the star is the object with the highest connectivity of the cluster. This algorithm relates sense representations in an analogous manner to the way in which syntactic or discourse relations link textual elements. To perform the clustering, this algorithm needs a similarity measure between objects (senses, in this case) and a minimum similarity threshold (β_0). In this prototype, the cosine similarity measure is used to compare senses.
Matching function: The matching function associates a three-component vector to a cluster g according to the textual context T as follows.

$$matching\text{-}function(g, T) = \left(|nouns(g)|, \frac{\sum_i \min(\bar{g}_i, T_i)}{\min(\sum_i \bar{g}_i, \sum_i T_i)}, -\sum_{s \in g} number(s) \right) \quad (1)$$

In this definition, $nouns(g)$ denotes the set of nouns associated to senses in g, $\bar{g}$ is the centroid of g, and $number(s)$ is the WordNet ordinal number of sense s (according to its corresponding noun). It is worth mentioning that T is represented in the sense VSM. The score assigned to each cluster considers

the number of nouns it has associated, its overlapping with the context and the WordNet sense frequency of its senses.

Filtering function: To perform the filtering, this function firstly sorts the clusters considering a lexicographic ordering of its associated matching-vectors obtained with (1). The goal of this function is to select clusters covering the set of nouns N by using the previous order. Thus, a cluster g is selected if it contains at least one sense of an uncovered noun and its senses corresponding to covered nouns are included in the already-selected clusters. If g does not contain any sense of uncovered nouns it is discarded. Otherwise, g is inserted into a queue Q. Finally, if the selected clusters do not cover N, clusters in Q adding senses of uncovered nouns are chosen until the cover is reached.

Stopping criterion: As the purpose of this algorithm is to disambiguate the set N, the cardinal of the set of senses obtained from the selected clusters in the filtering process must be equal to $|N|$. But it is not always possible with a single iteration. In order to obtain only one sense per noun, succesive clustering are required. Fine-grained clustering can be generated by increasing the minimum similarity threshold. Equation (2) gives the used threshold definition for the i-th iteration.

$$\beta_0(i) = \begin{cases} percentile(90, sim(S)) & \text{if } i = 0, \\ \min\limits_{q \in \{0,5,10\}} \{\beta = percentile(90+q, sim(S)) | \beta > \beta_0(i-1)\} & \text{otherwise.} \end{cases} \quad (2)$$

In this equation, $percentile(p, sim(S))$ represents the p-th percentile value of the set $sim(S) = \{cos(s_i, s_j) | s_i, s_j \in S, i \neq j\} \cup \{1\}$. The idea of this definition is to select at each time a threshold value that allows the arrangement of strong cohesive clusters. The selection is done from the set of pairwise similarities, trying to pick out a high percentile value. Hence, the stopping criterion is $|S| = |N|$ or $\beta_0(i+1) = 1$.

3.1 A Disambiguation Example

Figure 1 illustrates graphically the disambiguation of nouns in sentence "*The* ***competition*** *gave* ***evidence*** *of the* ***athlete's skills***". In this example, we consider N to be the set of lemmas of nouns in the sentence (bold-faced words), and the textual context T composed by these nouns plus the verb *give* (the rest of words are not included because they are meaningless). The correct senses of these nouns are *competition*#2, *evidence*#2, *athlete*#1 and *skill*#1.

The disambiguation algorithm firstly clusters the set of noun senses S = { *competition*#1, *competition*#2, *competition*#3, *competition*#4, *evidence*#1, *evidence*#2, *evidence*#3, *athlete*#1, *skill*#1, *skill*#2 }, using the initial $\beta_0 = 0.056$ (the 90th-percentile of the similarities between the senses). The boxes in the figure represent the obtained clusters, which are sorted by the matching scores (vectors under the boxes). As we can see, the first cluster comprises the single noun sense of *athlete*, *competition*#2, which is the sense refering to an athletic competition, and *skill*#1, which concerns to an ability acquired by training. Note that overlapped clusters are obtained. It can be also appreciated

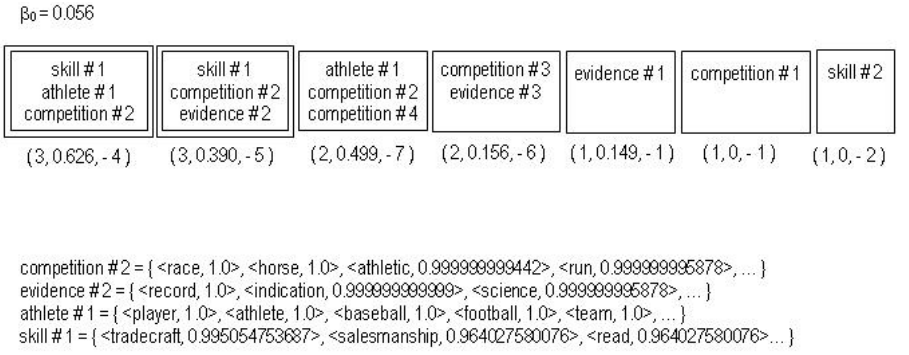


Fig. 1. Disambiguation of nouns in "*The competition gave evidence of the athlete's skills*"

that *competition*#2 and *competition*#4 form a cluster together with *athlete* due to its strong relation, although *competition*#4 is not a correct sense for the sentence.

Initially, the filtering function includes in the selection the first cluster. As the second cluster contains one sense of the uncovered noun *evidence* and all other senses are included in the first cluster, it is selected too. No other cluster is selected because all nouns are covered. In the figure, doubly-boxed clusters depict the selected ones by the filtering function. In this case, only one iteration is enough to provide the disambiguated nouns, because each noun has a unique sense in the set of selected clusters.

In addition, we have included in the bottom of the figure a portion of the representation of each resulting sense.

4 Experiments

In order to evaluate our approach, we use the SemCor corpus [11] and the traditional measures of *Precision*, *Recall*, *Coverage* and *F-measure*. This corpus comprises 190 documents containing 88026 noun occurrences, of which 69549 correspond to polysemous nouns.

In our experiments, the disambiguation process was carried out at the sentence level, assuming one sense per sentence. For each sentence, the set of its nouns is disambiguated considering all content words of the sentence as the textual context. To represent noun senses, topic signatures from the lexical relations of WordNet are built as follows. A topic signature for a sense includes all its hyponyms, its directly related terms and their glosses. To weight signatures terms, the tf-idf statistics is used, considering one collection for each noun, and its senses instead of documents.

Table 1 summarizes the overall recall split according to the SemCor categories. The columns include results for polysemous nouns only and for polysemous and monosemous nouns combined. In this table we have omitted the values

of *Precision*, *Coverage* and *F-measure* because all nouns are disambiguated by the algorithm, i.e. *Precision* and *F-measure* values coincide with *Recall*, and a 100 % of *Coverage* is achieved.

Table 1. WSD performance in SemCor categories

Categories	Polysemous nouns	All nouns
A. Press: reportage	0.606	0.683
C. Press: reportage	0.504	0.602
L. Mystery & detective fiction	0.498	0.589
F. Popular lore	0.482	0.604
P. Romance & love story	0.480	0.581
H. Miscellaneous	0.479	0.590
M. Science fiction	0.479	0.587
B. Press: editorial	0.476	0.599
K. General fiction	0.476	0.580
E. Skill & Hobbies	0.473	0.586
G. Belles letters, biography, essays	0.462	0.563
R. Humor	0.461	0.576
N. Adventure & western fiction	0.452	0.552
J. Learned	0.444	0.571
D. Religion	0.388	0.494
Brown 1	0.475	0.588
Brown 2	0.467	0.576
Whole SemCor	0.472	0.582

As shown in Table 1, our algorithm performs the best in *Press: reportage* category, and the worst in *Religion*. In all other categories the recall values are similar. Thus, it seems that the performance is not affected with different knowledge domains.

Since WordNet has been criticized for its lack of relations between topically related concepts, we evaluate the use of topic signatures developed by the Ixa Research Group[3], which attempts to overcome this drawback. These signatures were built by acquiring examples automatically from the Web with the monosemous relatives method for each WordNet nominal sense. In this case, the signature terms are those occurring in the retrieved snippets, and its weights are computed using the *tf-idf* statistics in a similar way.

Table 2 shows the results obtained in the disambiguation of all noun occurrences by using both topic signatures separatelly. The experiment was done using the 22 documents of Brown 1 belonging to categories from A to E.

Surprisingly, topic signatures built using only WordNet information outperform the Web-based ones. We suspect it is due to the fact that together with topically related concepts noisy terms are introduced in the signatures. Note also that the disambiguation of all noun occurrences was not possible using the Web-based topic signatures because some noun senses lack signatures.

[3] http://ixa.si.ehu.es/Ixa/

Table 2. Results using different topic signatures

Signatures	Polysemous nouns				All nouns			
	Recall	Precision	F	Coverage	Recall	Precision	F	Coverage
Based only on WordNet	0.501	0.501	0.501	100 %	0.603	0.603	0.603	100 %
Web-based	0.433	0.461	0.447	93.8 %	0.536	0.565	0.550	94.9 %

Finally, we compare our method with four knowledge-driven WSD algorithms: Conceptual density [2], UNED method [5], the Lesk method [8] and the Specification marks with voting heuristics [13]. Table 3 includes the recall values obtained over the whole SemCor corpus considering polysemous nouns only. Last column indicates whether the method obtained a full coverage or not.

Table 3. Overall performance

WSD method	Recall	Full coverage
Conceptual density	0.220	not
Lesk	0.274	not
UNED method	0.313	not
Specification marks	0.391	yes
Our method	**0.472**	**yes**

It can be appreciated that our approach improves all other methods in at least a 20 % of recall. It is important to notice that our method also ensures a full coverage.

5 Conclusions

In this paper a framework for the disambiguation of nouns has been introduced. Its novelty resides in the use of clustering as a natural way to connect semantically related word senses. Different knowledge-driven WSD methods can be obtained from it by specifying a representation for senses, a clustering algorithm, a matching and a filtering functions, and a stopping criterion. Some existing WSD methods can be seen as instances of this framework.

Most existing approaches attempt to disambiguate a target word in the context of its surrounding words using a particular taxonomical relation. Instead, we disambiguate a set of related words at once using a given textual context. Besides, we allow a flexible sense representation, with which more semantic information can be attached to the disambiguation process.

A first prototype algorithm for the framework was also introduced. It relies on both topic signatures built from WordNet and the Extended Star clustering algorithm. The way this clustering algorithm relates sense representations resembles the manner in which syntactic or discourse relations link textual components.

This algorithm was compared with other knowledge-driven disambiguation methods over the whole SemCor corpus. The experimental results show that our algorithm obtains better recall values, while achieving a 100 % of coverage.

Though in this work we treat only the disambiguation of nouns, the approach can be extended to consider other word categories. As further work, we plan to generate new algorithms from the framework, using other clustering algorithms and varying the textual contexts, to explore its impact in the disambiguation task.

References

1. Agirre, E., López, O.: Clustering wordnet word senses. In: Proceedings of the Conference on Recent Advances on Natural Language Processing. Bulgary (2003) 121–130
2. Agirre, E., Rigau, G.: Word Sense Disambiguation Using Conceptual Density. In: Proceedings of the 16th Conference on Computational Linguistic, Vol. 1. Denmark (1996) 16–22
3. Bordag, S.: Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. Accepted to the 11st Conference of the European Chapter of the Association for Computational Linguistic. Italy (2006)
4. Cowie, J., Guthrie, J. A., Guthrie, L.: Lexical disambiguation using simulated annealing. In: Proceedings of the 14th International Conference on Computational Linguistics, Vol. 1. France (1992) 359–365
5. Fernández-Amorós, D., Gonzalo, J., Verdejo, F.: The Role of Conceptual Relations in Word Sense Disambiguation. In: Proceedings of the 6th International Workshop on Applications of Natural Language for Information Systems. Spain (2001) 87–98
6. Gil-García, R., Badía-Contelles, J.M., Pons-Porrata, A.: Extended Star Clustering Algorithm. Progress in Pattern Recognition, Speech and Image Analysis. Lecture Notes on Computer Sciences, Vol. 2905. Springer-Verlag (2003) 480-487
7. Ide, N., Véronis, J.: Word Sense Disambiguation: The State of the Art. Computational Linguistics 24:1 (1998) 1–40
8. Lesk, M.: Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation. Canada (1986) 24–26
9. Lin, C.-Y., Hovy, E.: The Automated Acquisition of Topic Signatures for Text Summarization. In: Proceedings of the COLING Conference. France (2000) 495–501
10. Mihalcea, R., Moldovan, D.I.: EZ. WordNet: Principles for Automatic Generation of a Coarse Grained WordNet. In: Proceedings of the FLAIRS Conference. Florida (2001) 454–458
11. Miller, G. A., Leacock, C., Randee, T., Bunker, R.: A Semantic Concordance. In: Proceedings of the 3rd DARPA Workshop on Human Language Technology. New Jersey (1993) 303–308
12. Miller, G.: WordNet: A Lexical Database for English. Communications of the ACM 38:11 (1995) 39–41
13. Montoyo, A., Suárez, A., Rigau, G., Palomar, M.: Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods. Journal of Artificial Intelligence Research 23 (2005) 299–330

14. Niu, C., Li, W., Srihari, R.K., Li, H., Crist, L.: Context Clustering for Word Sense Disambiguation Based on Modeling Pairwise Context Similarities. In: SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text. Spain (2004) 187–190
15. Patwardhan, S., Banerjee, S., Pedersen, T.: Using Measures of Semantic Relatedness for Word Sense Disambiguation. In: Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing. Mexico (2003) 16–22
16. Pedersen, T., Purandare, A., Kulkarni, A.: Name Discrimination by Clustering Similar Contexts. In: Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing. Mexico (2005) 226–237
17. Rosso, P., Masulli, F., Buscaldi, D., Pla, F., Molina, A.: Automatic Noun Sense Disambiguation. In: Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing. Mexico (2003) 16–22
18. Rosso, P., Montes-y-Gómez, M., Buscaldi, B., Pancardo-Rodríguez, A., Villaseñor-Pineda, L.: Two Web-Based Approaches for Noun Sense Disambiguation. In: Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing. Mexico (2005) 267–279
19. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Information Retrieval. Journal of the American Society for Information Science 18:11 (1975) 613–620
20. Udani, G., Dave, S., Davis, A., Sibley, T.: Noun Sense Induction Using Web Search Results. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Brazil (2005) 657–658

Comparing Two Markov Methods for Part-of-Speech Tagging of Portuguese

Fábio N. Kepler and Marcelo Finger

Institute of Mathematics and Statistics, University of São Paulo (USP)
{kepler, mfinger}@ime.usp.br

Abstract. There is a wide variety of statistical methods applied to Part-of-Speech (PoS) tagging, that associate words in a text to their corresponding PoS. The majority of those methods analyse a fixed, small neighborhood of words imposing some form of Markov restriction. In this work we implement and compare a fixed length hidden Markov model (HMM) with a variable length Markov chain (VLMC); the latter is, in principle, capable of detecting long distance dependencies. We show that the VLMC model performs better in terms of accuracy and almost equally in terms of tagging time, also doing very well in training time. However, the VLMC method actually fails to capture really long distance dependencies, and we analyse the reasons for such behaviour.

1 Introduction

An empiric approach to Natural Language Processing (NLP) suggests that we can learn the complicated and extensive structure of language by searching common patterns in its use, identified via Probability Theory. This leads to probabilistic models of linguistic phenomena whose parameters are induced by the application of statistical and machine learning methods.

This work concentrates on *part-of-speech analysis*, that is, attributing to words in context their correspondent part-of-speech (PoS) tag. Several computational linguistic tasks benefit from such analysis, such as parsing, automatic translation, grammar correction and information extraction.

Many words have more than one possible PoS tag, which is disambiguated by the context in which these words occur. However, even with contextual information some words remain ambiguous.

Several computational approaches use statistical methods to model relations between words, tags and contexts, which are normally computed by employing some form of supervised learning from a manually tagged corpus. Other approaches exist in literature like neural networks and symbolic methods.

A traditionally used method is the Hidden Markov Model (HMM) [1], that tags a word generally by analysing the tags of the two previous words. This method works quite well, classifying correctly more than 93% of the words of a Portuguese corpus. This relatively great accuracy is due to the fact that most contextual dependencies is short distanced, so that just the two previous tags

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 482–491, 2006.

can determine the tag of a word. But considering that the tagging accuracy of several existing limited-context methods have practically converged to a value inferior to 100% (∼97%), an extra portion of efficiency could be reached developing methods that consider longer distance dependencies. However, due to data sparseness, contexts with size greater than two can hardly improve accuracy on these methods, since several contexts occur only once in the corpus, and therefore cannot capture any dependency. This can be seen even with contexts of size two, as shown in the results (Section 4). Moreover, the time necessary to train a HMM model, that is, to count and to calculate the probabilities of tags in context, is usually too large, with an exponential growth on the number of words considered in the context. The other method we study is Variable Length Markov Chains (VLMC), and we use it to try to capture *variable* long distance dependecies, thus overcoming these problems and reaching that extra efficiency.

This work presents a comparison between two methods applied to PoS tagging of Portuguese: the classically used HMM method, and the not so known VLMC method. It also presents a study about the treatment of short, average, and long distance dependencies in the Brazilian Portuguese when using VLMC. We obtained accuracy results very near the best ones reported for Portuguese, and also with training and execution times well below the existing ones in literature.

The article is organized as the following. Section 2 briefly presents the application of VLMC to the PoS tagging problem. Section 3 describes the taggers' implementation. The obtained results are analysed in Section 4, and a comparison with similar results in literature is shown in Section 5. Conclusions about the application of the methods are then discussed in Section 6.

2 Variable Length Markov Chains

The idea is to allow the memory of a Markov chain to have variable length, depending on the observed past values. We will explain the concept of Variable Length Markov Chains considering the PoS tagging context. For a formal description see [2,3].

Consider a Markov chain with a finite, large order k. Let l_i be a tag, and $l_{i-k,i-1}$ be the tags preceding l_i. The idea of a variable length memory can be seen as a cut of irrelevant states from the $l_{i-k,i-1}$ history. We call the set of these states the *context* of l_i.

Given a tag l_i, its context $l_{i-h,i-1}$, $h \leq k$, is given by the *context function* $c(l_{i-k,i-1})$. The states that determine the probabilities of the VLMC are given by the values of the context functions of the tags. These states are represented as a tree.

A *context tree* is a tree with a root node in the top, from which ramifications go down, such that each internal node has at most $|\mathcal{L}|$ sons, where $\mathcal{L}$ is the tagset. Each value of a context function $c(\cdot)$ is represented as a branch of such tree. For example, the context given by $c(l_{i-k,i-1})$ is represented as a branch

whose sub-branch in the top is determined by l_{i-1}, the next sub-branch by l_{i-2}, and so on, until the leaf, determined by l_{i-h}.

The parameters of a VLMC are the underlying functions $c(\cdot)$ and their probabilities. To obtain these parameters we use a version of the context algorithm of Rissanen [4]. First, it builds a big context tree, using a training corpus. For a tag l_i, its maximal history $l_{i-k,i-1}$ is placed as a branch in the tree. Then, the algorithm uses a pruning function considering a local decision criterion. This pruning cuts off the irrelevant states from the tags' histories. For each leaf u of the context tree, and branch v that goes from the root to the upper node of u, u is pruned from the tree if

$$\Delta_{vu} = \sum_{l \in \mathcal{L}} P(l|vu) \log \left(\frac{P(l|vu)}{P(l|v)} \right) C(vu) < K, \tag{1}$$

where $C(vu)$ is the number of occurrences of the sequence vu in the training corpus, and K is a threshold value, called the *cut value* of the context tree.

If the probability of a tag does not change much between considering the entire branch together with the leaf (all past history) and considering only the branch (the history without the oldest tag), then the leaf does not need to be considered, and can be removed from the tree.

3 Implementation

We have implemented two PoS taggers, one based on VLMC and one based on HMM.

The HMM tagger uses second order Markov models — that is, the tag of a word depends on the word itself and on the two previous tags — where the states represent the tags and the observations represent the words. The transition probabilities depend on the states, in this case pairs of tags, and the outcome probabilities depend on the destination state. Formally, we wish to find the best sequence of tags $l_1 \dots l_T$ for a given sequence of words $w_1 \dots w_T$ of size T, that is,

$$\arg\max_{l_1 \dots l_T} \left[\prod_{i=1}^{T} P(l_i|l_{i-1}, l_{i-2}) P(w_i|l_i) \right] \tag{2}$$

where l_0 and l_{-1} are sentence starting markers.

The probabilities are estimated from a tagged corpus. We use maximum likelihood probabilities $\hat{P}$, which are derived from the relative frequencies of words and sequences of tags. For example, the probability of a tag l_3 given the sequence l_1, l_2 is equal to the number of times the three tags occur in sequence (l_1, l_2, l_3) in the corpus divided by the number of times only the two previous tags occur in sequence (l_1, l_2). That is, $\hat{P}(l_3|l_1, l_2) = \frac{C(l_1,l_2,l_3)}{C(l_1,l_2)}$.

In the VLMC tagger the sequences of tags obtained from the training corpus are used to feed the context tree, which after pruned defines the context functions. This way, for a given sequence of tags $l_{i-k,i-1}$, the probability that the next tag is l_i is given by $P(l_i|c(l_{i-k,i-1}))$.

We decided to use the value of K given by an equation defined empirically:

$$K = \frac{\log(n)}{\log(|\mathcal{L}|)} \cdot \frac{n}{|\mathcal{S}|}, \tag{3}$$

where $|\mathcal{L}|$ is the size of the tagset, $|\mathcal{S}|$ the number of sentences, and n the number of words of the training corpus. For the whole training corpus (775,602 words), K was equal to 54.6615. The results showed in Section 4 were obtained with this equation.

3.1 Treating Unknown Words

When a text is tagged, words new to the training corpus are found. Thus, in order to tag them more correctly as possible, a special treatment is needed.

We used two complementary standard methods. The first consists in restricting the possible tags for an unknown word, eliminating closed tags. *Closed tags*, like articles and conjunctions, are those tags that are only assigned to a limited number of different words. Yet other tags like verbs and nouns can be assigned to a great number of distinct words, and therefore are called *open tags*. Though this distinction can be easily made by hand it is not passed to the taggers. Instead, they are given a threshold that is used to build these two sets of tags after training: if a tag occurs with too many different words, it is an open tag; otherwise, it is a closed tag.

In the second method the word's morphology is analyzed. We use a simplificated suffix analysis[1]: a tree is built with the suffixes of all words from the training corpus that have an open tag; next, this tree is normalized, creating probability distributions; then, for an unknown word, a search for its greatest suffix existing in the tree is made, returning the probabilities of the possible tags. We consider as the suffix of a word used to build the tree the last half of the word. Moreover, we check to see if the first letter of a word is uppercase and if so, if this word is not sentence starting, we assign to it a higher probability of being a proper noun. Note that no external dictionary is used.

4 Tests and Results

Both taggers were implemented using `C++` and `STL` (*Standard Template Library*), and were compiled with `g++` version 3.3.4. The tests were made on a machine equipped with one Intel Pentium 4 processor of 3 GHz, and 1 GB of RAM.

The taggers were trained and tested with the *Tycho Brahe* corpus [5], which uses a set of 383 tags and contains various texts from historical Portuguese manually tagged, in a total of 1,035,593 words. These words were splitted into a training corpus containing 775,602 words, and a testing corpus containing 259,991 words.

[1] The term suffix we use means "final sequence of characters of a word", what is not necessarily a linguistically significant suffix.

We executed sets of tests varying the size of the training corpus, choosing $5\%, 10\%, \ldots, 95\%, 100\%$ of its sentences and executing 10 times with each one of these sizes (randomizing the sentences each time), but always using the testing corpus without modifications.

Figure 1 shows accuracy[2] results for both taggers. It can be seen that the VLMC tagger has consistently a greater accuracy than the HMM tagger's one; for the whole corpus, 95.51% over 93.48%. The VLMC tagger's curve grows quicker than the HMM tagger's one. Based on this information we can say that, even increasing the number of words in the training corpus, the HMM tagger will not reach the accuracy of the VLMC tagger.

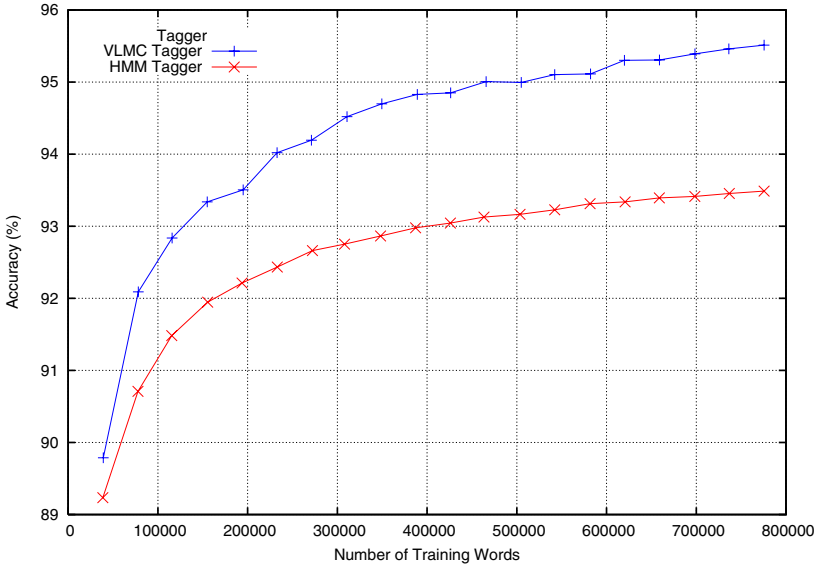


Fig. 1. HMM tagger's and VLMC tagger's accuracy

In the graph of the Figure 2 three curves are shown, representing the total accuracy of the VLMC tagger, its accuracy for known words, and its accuracy for unknown words. When using the entire training corpus, 69.53% of the unknown words are correctly tagged. With only 5% of the training corpus, the average accuracy for unknown words was only 63.97%, a difference of more than 5%. Considering the fact that the number of unknown words is greater when using a smaller training corpus, this difference represents approximately 4% of the testing corpus, which means more than $10,650$ tagging mistakes.

With respect to known words, the average accuracy with 5% of the training corpus was 94.20%, and with the entire corpus 96.39%. This difference is approximately $3,872$ words, about 1.49% of the testing corpus.

[2] By accuracy we mean the proportion of words from the testing corpus to which the tagger assigns the correct tag.

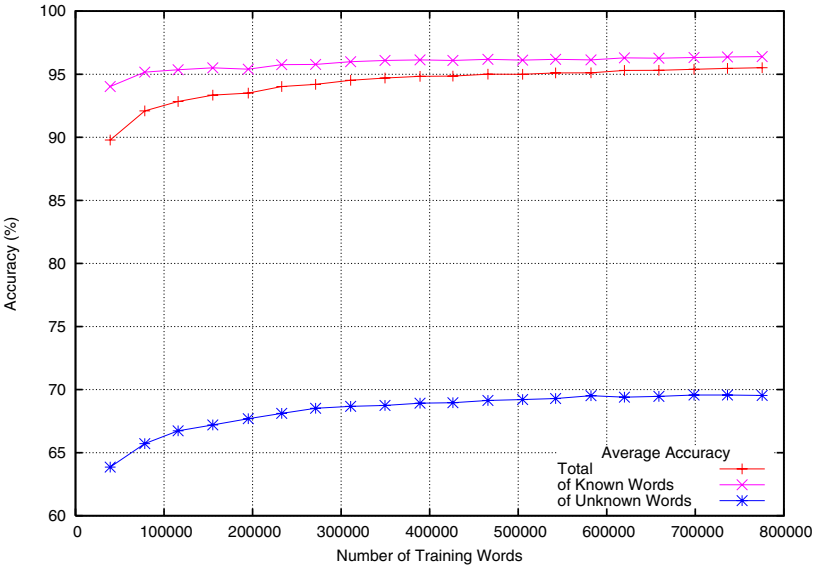


Fig. 2. VLMC tagger's accuracy for known and unknown words

In the HMM tagger, the percentage of unknown words correctly tagged varied from 62.37% with 5% of the training corpus, to 68.81% with the entire corpus; a difference of 6.5%, which in the VLMC tagger was of 5%. This can be explained by the fact that the HMM tagger considers very small contexts, and therefore it ends out leaving the choice of the best tag for an unknown word only to the suffix tree. This is why an increase in the number of training words improves the accuracy for the unknown words.

The fact that the HMM tagger considers very small context also explains the interesting result obtained for the known words. The variation in the accuracy according to the size of the training corpus was small: 93.62% with 5% of it, and 94.32% with the entire corpus. A difference of only 0.7%, against 2.19% from the VLMC tagger. That is, because the HMM tagger considers small contexts, tagging the known words is little influenced by the tagging of the unknown words. This shows that, at the same time that the VLMC tagger treats better unknown words, since it uses bigger contexts as aid, it is also more sensible to eventual tagging mistakes, and may wrongly tag a word by influence of the context.

Figure 3 shows the curves of the average training and tagging times taken by the taggers with respect to the number of words used to train them. Both training and tagging times for the HMM tagger are slightly smaller than the ones for the VLMC tagger. But in fact, considering the total execution time of a tagger as the sum of its training and tagging times (and some constant overhead time for system operations), the curve of the HMM tagger's execution time has a correlation of 0.9956, and the VLMC tagger's one of 0.9969. This means that

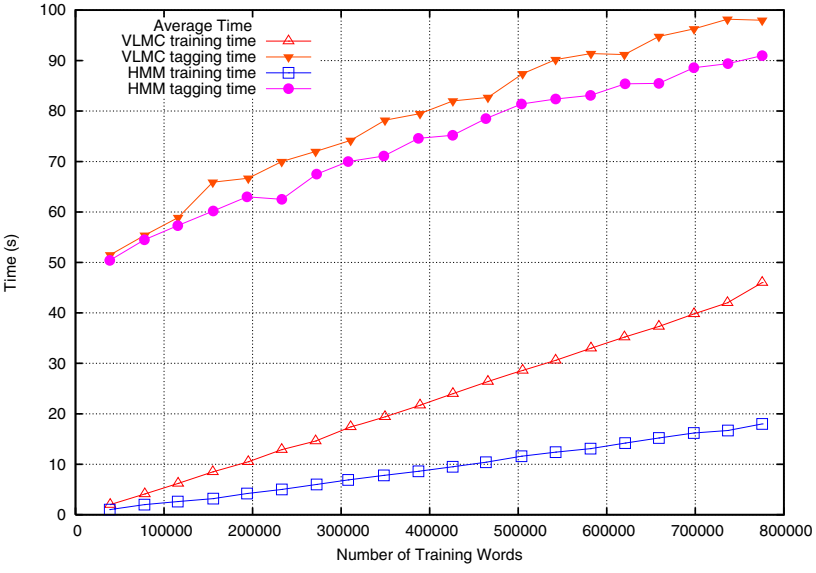


Fig. 3. HMM tagger's and VLMC tagger's average training and tagging times

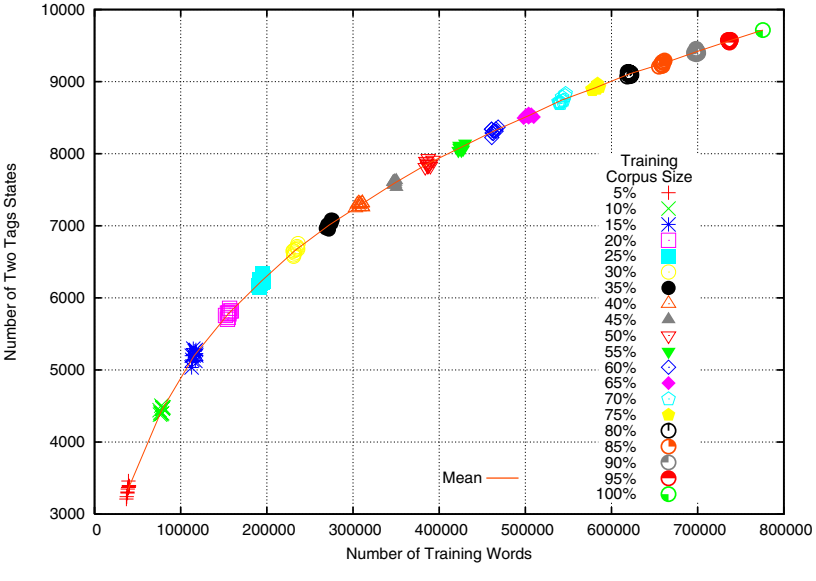


Fig. 4. Number of states of two tags obtained by the HMM tagger with respect to the number of words in training

both taggers have linear complexity over time, and so both are equally efficient with respect to execution time.

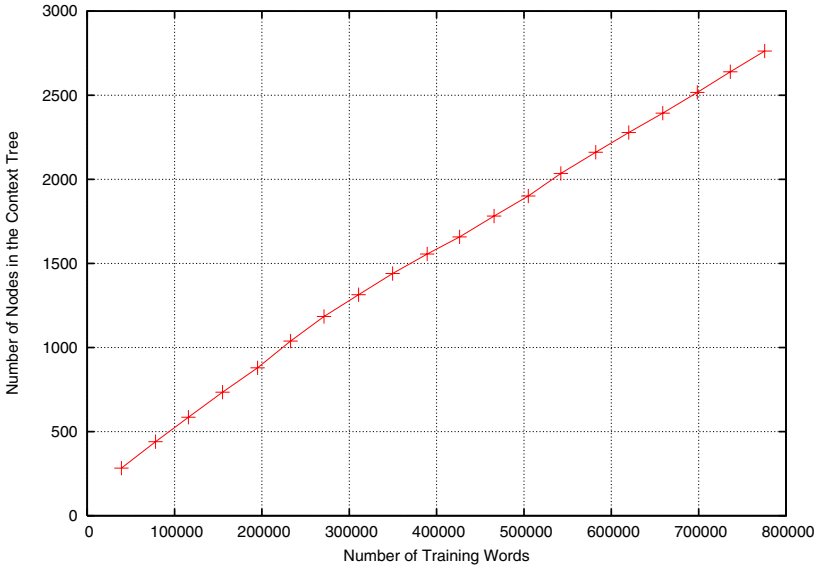


Fig. 5. Growth of the number of nodes in the VLMC tagger's context tree

However, note that the HMM tagger's training time is relatively low. The explanation is that we did not make any training iteration for it, as normally is done. On the contrary, we just learned the probabilities through the training corpus. This was done because the time needed for the training process was too great, around 4622 seconds (37 times greater than without training), and the accuracy did not improve as much as expected. Moreover, works like the ones of Church [6] and DeRose [7] also collect statistical information from a tagged corpus instead of using the process of training for HMM.

Figure 4 shows the number of states of two tags built by the HMM tagger during training. Figure 5 shows the growth of the number of nodes in the VLMC tagger's context tree. Comparing both figures, it can be seen that the number of states in the HMM tagger is more than three times larger than the number of nodes in the VLMC tagger, showing that many sequences of two tags from the training corpus are consequence of sparse data, and do not add much knowledge relevant for the tagging.

Other results were also generated, but these are omitted due to space restrictions. The interested reader can consult [reference withold].

5 Related Works

A currently well known tagger is the one of Brill [8], which is based on transformation rules and achieves around 95.4% of accuracy in the Wall Street Journal (WSJ) [9] English corpus. For the Portuguese, Chacur and Finger [10] proposed

and implemented a variant of the Brill's method, and obtained reasonable results. After that Finger [11] used some optimization techniques and obtained better results, around 95.43% of accuracy on the Tycho Brahe [5] corpus.

Among the taggers that use statistical models Ratnaparkhi [12] implements one based on Maximum Entropy, which obtains 96.6% of accuracy on the WSJ corpus. Brants [13] implements a tagger based on HMM that achieves an accuracy of 96.7%. Recently, Toutanova [14] showed a tagger based on Ciclic Dependendy Network, which obtains 97.24% of accuracy, claiming this is the state of the art for English. For Portuguese, Aires [15] adapts various English taggers and shows their results for Portuguese. The best one achieves 90.25% of accuracy, and is obtained by the adaptation of the Ratnaparkhi's Maximum Entropy tagger.

6 Conclusions

We built two PoS taggers for Portuguese, one based on the traditionally used HMM, and one based on VLMC, a recent theoretic statistical model.

With an accuracy of 95.51% with the VLMC tagger we obtained a result very close to the best ones reported for Portuguese [10,15,11]. Also, the time spent in training with more than $775,000$ words and in tagging almost $260,000$ ($1,035,593$ in total) is very fast, though we cannot compare it to other taggers in literature since this result is normally not presented (even so, we report that the VLMC tagger tooks only 157 seconds to train and tag with the million words stated above, running on cheap machine).

When instructing the VLMC tagger to consider longer contexts, it was not able to detect many long distance dependencies. Moreover, instructing it to consider not so long contexts have improved the performance (in terms of accuracy). So we conclude that, when having less long contexts available, the tagger chooses short and recent contexts, what improves the performance and shows that there are long contexts that decay it.

Though Variable Length Markov Chains do not capture very long contexts, they perform consistently better for part-of-speech analysis than the classicaly applied theory of fixed order Markov Chains. We give results that allow us to observe limitations and advantages of the application of statistical models based on VLMC: they learn various short and average distance fixed contexts ($d \leq 6$), but they do not have generalizating capacity to learn linguistic phenomena occuring in variable contexts and of longer distance. Future research in statistical linguistics regarding long range dependencies should concentrate in other ways of solving this limitation.

References

1. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. In: Proceedings of the IEEE. Volume 77 of no. 2. (1989) 257–285
2. Bühlmann, P., Wyner, A.J.: Variable length markov chains. Annals of Statistics **27**(2) (1999) 480–513

3. Mächler, M., Bühlmann, P.: Variable length markov chains: Methodology, computing and software. Research Report 104, Eidgenossische Technische Hochschule (ETH), CH-8091 Zürich, Switzerland (2002) Seminar fur Statistik.
4. Rissanen, J.: A universal data compression system. IEEE Trans. Inform. Theory **IT-29** (1983) 656 – 664
5. IEL-UNICAMP and IME-USP: Corpus Anotado do Português Histórico Tycho Brahe. (2005) Acessado em 2005.
6. Church, K.W.: A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the second conference on Applied natural language processing, Austin, Texas, Association for Computational Linguistics (1988) 136–143
7. DeRose, S.J.: Grammatical category disambiguation by statistical optimization. Computational Linguistics **14** (1988) 31–39
8. Brill, E.: Unsupervised learning of disambiguation rules for part of speech tagging. In Yarovsky, D., Church, K., eds.: Proceedings of the Third Workshop on Very Large Corpora, Somerset, New Jersey, Association for Computational Linguistics (1995) 1–13
9. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics **19**(2) (1994) 313–330
10. Alves, C.D.C., Finger, M.: Etiquetagem do português clássico baseada em córpora. In: Proceedings of IV Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR99), Évora, Portugal (1999) 21–22
11. Finger, M.: Técnicas de otimização da precisão empregadas no etiquetador Tycho Brahe. In: Proceedings of V Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR2000), Atibaia, Brazil (2000) 19–22
12. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Proceedings of the Empirical Methods in Natural Language Processing Conference, University of Pennsylvania (1996)
13. Brants, T.: Tnt – a statistical part-of-speech tagger. In: Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000), Seattle, WA (2000)
14. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL 2003. (2003) 252–259
15. Aires, R.V.X.: Implementação, adaptação, combinação e avaliação de etiquetadores para o português do brasil. Dissertação de mestrado, Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo - Campus São Carlos (2000)

Shallow Parsing Based on Comma Values

Diego Garat

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Herrera y Reissig 565
11300 Montevideo, Uruguay
dgarat@fing.edu.uy

Abstract. In the belief that punctuation can aid in the process of sentence structure analysis, our work focuses on a prior assignment of values to commas in Spanish texts. Supervised machine learning techniques are applied for learning commas classifiers, taking as input attributes positional information and part of speech tags. One of these comma classifiers and a rule-based analyzer are combined in order to recognize and label text structures. The prior assignment of values to commas allowed the simplification of recognition rules, with very encouraging results.

Keywords: punctuation, shallow parsing, decision trees, boosting.

1 Introduction

Although punctuation, specifically commas, has widely been considered as pausing and intonation marks for reading, this is clearly not its unique function: punctuation has a key role in the interpretation of texts. This role has been recognized in various approaches to structural analysis of texts. Jones [4] carried out an experiment aimed to determine the contribution of punctuation in symbolic analysis. The results indicate a dramatic decrease in ambiguity (i.e. the number of parse trees) for complex sentences when rules take account of punctuation symbols.

Briscoe and Carroll [2] integrate punctuation into a probabilistic LR-parser. Also in this case, the test driven by comparing the results of parsing with and without rules for punctuation assesses the contribution of punctuation to the task of finding the correct analysis. Punctuation has also been analyzed from a generation perspective. White [9] proposes a stratified account for the generation of punctuation marks, and sketches its integration in the Meaning-Text framework.

In particular, comma values in English texts have been analyzed by Bayraktar, Say and Akman [1]. They present a classification of its uses, and relate comma values to syntactic patterns obtained from the Penn Treebank.

Van Delden y Gómez [8] took a different approach: they applied finite state techniques to determine the syntactic role of commas in English texts. They manually constructed a set of automata and, with a co-occurrence matrix, they determined the possible values of commas.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 492–501, 2006.

Our work goes a step further: we *automatically* construct a comma evaluator using machine learning techniques. The comma classifier is learned from manually tagged examples, taking part of speech tags and positional information as input attributes. The resulting classifier is combined with a rule-base analyzer as an aid for structural assignment of sentences. Our work also departs from previous works since it has been carried out for Spanish; thus, we had to study the usage of commas for this language to establish a classification suitable for our task.

In section 2, we present the classification proposed for commas in Spanish texts. Section 3 develops the details of the experiments done to obtain the comma classifier and the results obtained. The rule-based parser is presented in section 4. Finally, conclusions are discussed in section 5.

2 Classification

The Spanish Real Academy (RAE) distinguishes almost 20 different linguistic structural usages of commas in Spanish [6]. Six of these usages are for incises, others are for series, others are disambiguating commas.

Instead, we have divided the comma usages in five main groups: incise, series, modifiers, connective, bipolar and others, totalizing 9 different labels (inside the incise and series classes we distinguished different subtypes). This set is a trade-off between RAE's classification and practical considerations. Although it may seem that they are too many labels, preliminary tests with smaller sets give worse results, maybe because some subtypes –for instance, discoursive incises– have less and simpler patterns to be learned.

Table 1. Classification results

Type	**Cant**	**%**
Incise Begin (CMIB)	1757	32,8
Incise End (CMIE)	948	17,7
Incise Discourse(CMID)	141	2,6
Series Other (CMSO)	832	15,5
Series Propos. (CMSP)	790	14,8
Modifier (CMMO)	542	10,1
Connector (CMCO)	198	3,7
Bipolar (CMBI)	91	1,7
Others (CMO)	52	1,0
Total	**5351**	100,0

The corpus is composed of 77 press articles, from "Página/12" newspaper[1]. From these articles, more than 5000 examples are extracted. Because one comma may have more than one usage in a sentence, more than one label could possibly be assigned. In case of multivaluated commas, one example is created for each label of each comma. The distribution of the different classes is shown in table 1.

[1] http://www.pagina12web.com.ar

1. $^{[CMIB]}$A un lado del mostrador^{CMIE},^{CMIB} custodiadas por una bandera de colores brillantes que recuerda el lugar donde todo comenzó ^{CMIE}, las "fresas" siguen recordando las costumbres del pueblo que creó las recetas.

   $^{[CMIB]}$Aside of the counter^{CMIE},^{CMIB} guarded by a brilliant flag of colors that recalls the place where all began ^{CMIE}, the "strawberries" continue recalling the customs of the town that created the recipes.

2. No se sabe,^{CMIB} en realidad^{CMIE}, si lo que atrae es la promesa de galletas tibias,^{CMSO} el anisado de unas rosquitas,^{CMSO} la hogaza de pan dorada en su punto justo o todo eso junto.

   It is unknown, in fact, if what attracts is the promise of tepid crackers,^{CMSO} the aniseed of some *rosquitas*,^{CMSO} the mold of golden bread in its point or all that together.

3. Hasta hoy,^{CMMO} pasados diez meses,^{CMMO} no se sabe por qué mataron a Piazza,^{CMSP} pero todos los indicios van en dirección a un crimen policial.

   Until today,^{CMMO} after ten months,^{CMMO} it is unknown why they killed Piazza,^{CMSP} but all the clues point to a police crime.

4. Además,^{CMCO} dejó toda la impresión de mentir, ya que estudios posteriores...

   Besides,^{CMCO} it left all the impression of lying, since subsequent studies...

Fig. 1. Examples of labeled texts

Hereafter, we give a brief description of each class. Examples of labeled texts are shown in Figure 1.

Incises. Two distinct marks are used for incises commas: beginning (CMIB) and end incise comma (CMIE). Clearly, incises are not only delimited by a pair of commas: sometimes the beginning or the end of the sentence acts as a boundary. Also, the same comma may act as the mark of an incise beginning and another incise end (see sentences 1 and 2 in Figure 1).

Because the working corpus was obtained from online newspapers, we found useful to distinguish another type of incises: discursive ones (CMID). Generally, this type of commas precedes discursive verbs, such as say, tell, communicate, transmit, opine, think, etc.

Series. Two kinds of series commas are distinguished: the ones which act as a limit of propositional segments (CMSP) and the commas that delimit simple element series(CMSO). The later includes nominal, adjectival and adverbial series (see sentences 2 and 3 in Figure 1).

Modifiers. Commas classified as "modifier" (CMMO) appear after certain adverbs that usually modify the entire sentence, and not only some of its elements. For example: "until now", "after ten months", "in Paris", etc. They are generally placed at the beginning of the sentence. (see sentence 3 in Figure 1).

Connective. Connective commas (CMCO) are used after certain discourse connectors such as "finally", "also", etc. As modifiers, they are generally placed at the beginning of the sentence (see sentence 4 in Figure 1).

Bipolar. A comma classified as "bipolar" (CMBI) gives account of certain constructions (hypothetical, concessive) with two members of equal importance (despite the fact that one of them is usually marked as subordinated and the other as main clause in a traditional approach).

Others. Commas not considered in the previous categories are grouped under this category. This includes the following phenomena: transposition, elided verb and, of course, misusage.

3 Classifiers

3.1 Training Set

The classifier is intended to be used in between a POS tagger and a structure analyzer. Thus, the attributes to be considered should be obtained by the POS tags or calculated directly from them or from the text.

Initially, only a window of ten POS tags surrounding the comma was considered. This led to a classifier with an error rate of more than 47%. As a consequence, several attributes were created in the belief that they would help the learning process.

For example, when the label "discursive incise" was added to the categories set, we found useful to incorporate an attribute to signal the presence (or absence) of a discursive verb near the comma. If this kind of verb cannot be found near the comma, it seems quite difficult that it be labeled as "discursive incise".

Another example is the pattern attribute: it takes the value of a POS tag for an adjective (ADJ), a verb (VERB) or a noun (NOUN) that appears repeatedly after or before the comma, separated by others commas or a conjunction (CONJ). This attribute seems useful for detecting the presence of series of adjectives, verbs, proper names, etc., because generally, these series are of the form: "Pattern, ..., Pattern CONJ Pattern". Patterns can be constituted by more than one category, a determiner and a noun for example; if that is the case, a "representative" label is selected (in the previous example, it would be "noun").

In our final experiment, the attributes considered are: verbs that appear near the comma, minimum distance from the next or previous comma (measured as lexical units), possible patterns, presence of a discourse verb, an indication about being (or not) the first comma in the sentence and an indication about being the last one. The sentence limit category (SENT) is considered when there are no sufficient categories to complete the ten tokens length window. The POS tags are obtained using Freeling[3].

For example, given the sentence:

"Horario central" es una obra para cinco personajes, un poco loca, pero a mí me gusta deformar lo que aparenta ser normal.

"Central time" is a play for five characters, a little crazy, but I like to distort what seems to be normal.

the values of the first comma attributes are:

Connector: NO	Cat_{after}: NONE	Cat_{before}: CC
$Verb_{before}$: OTHER	$Verb_{after}$: OTHER	Distance: 4
Pattern: NONE	First: YES	Last: NO
Cat_{-5}: DIS	Cat_{-4}: NC	Cat_{-3}: SP
Cat_{-2}: DNP	Cat_{-1}: NC	Cat_{+1}: DIS
Cat_{+2}: RG	Cat_{+3}: AQS	Cat_{+4}: FC
Cat_{+5}: CC		

All the attributes values are determined exclusively by using the words of the sentence in which the comma is present, due to the local scope of this punctuation mark.

3.2 Training

For this work, we focused on two machine learning algorithms: decision trees [5] and Boostexter [7].

Decision trees have being used in multiple NLP tasks: POS tagging, sense disambiguation, cue phrase identification, etc. The advantage of this algorithm is that the output is "readable", i.e. it can be transformed into a set of rules easily interpretable by humans. In our first experiments, we applied Quinlan's C4.5 algorithm.

Due to the low level of precision acquired with this algorithm (explained in the next section), we decided to try Schapire's and Singer's Boostexter. Being a boosting algorithm, it combines several simple classifiers to obtain a more accurate one. Each classifier is trained over the set of examples that the previous ones have poorly behaved.

3.3 Results

We trained C4.5 using ten-cross-fold-validation; the precision rate was 60%. Instead, Boostexter performed slightly better: using 300 examples as the test set and the remaining examples as the training set, the precision increases to 66%. In both cases, commas labeled as CMBI and CMO are discarded due to their low frequency.

These rates seem quite disappointing, but if we consider the base error (taking the most common category, i.e. CMIB), the error percentage is lowered in approximately 30 and 40 points by C4.5 and Boostexter respectively.

As shown in Figure 2, the results are not equal for all categories. The classifiers behave poorly in commas valued as "propositional series" and better in "discourse incises". These results are probably due to the number of different

Table 2. F-measure of the classifiers

	Tree	Rules	Boostexter
CMIB	68,4	66,2	**72,6**
CMIE	51,0	57,7	**64,2**
CMID	**77,8**	**77,8**	72,7
CMSO	55,1	56,6	**61,3**
CMSP	54,8	48,3	**55,1**
CMCO	47,1	69,4	**76,9**
CMMO	52,0	48,3	**64,4**

syntactic patterns in which each type of comma appears. Bayraktar et al. observed the same phenomena in English texts: commas in series are less "stable" than others.

4 Syntactic Analyzer

Is the information given by the classifiers useful despite their error rate? Our experiments lead to an affirmative answer.

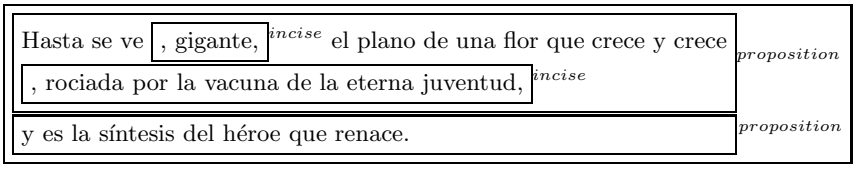
We used the classifier output by Boostexter in interaction with a rule-base analyzer to mark segments in texts enclosed by punctuation. Our objective is to take the maximum advantage of the information given by the classifier.

Three major types of structures are considered: simple series, propositional segments and incises. Initial modifiers and discourse connectives are treated as incises, because of their similar behavior in text structures.

For example, given the following text:

> Hasta se ve, gigante, el plano de una flor que crece y crece, rociada por la vacuna de la eterna juventud, y es la síntesis del héroe que renace.
>
> It is also seen, giant, the plane of a flower that grows and grows, poured by the eternal youth vaccine, and it is the synthesis of the hero that is born again.

the desired output is:



The analyzer is based on "Contextual Rules"[10], a rewriting formalism originally developed for recognition of propositions, but suitable for a large number of syntactic labeling tasks.

A contextual rule is an expression whose purpose is to identify and label portions of text. A portion of text gets labeled if it satisfies a condition that is a function of the same portion and, eventually, of a portion of text that precedes

it (left context) and/or of a portion of text that follows it (right context). A portion of text satisfies the condition that allows the labeling if it includes certain elements (words, punctuation marks, portions of texts previously labeled) in a specific order. The rule determines what are the elements that have to be present and the order between them. These elements have not always to be contiguous. Intercalated between them there can be other portions of text for which the rule only states the maximum size and a list of elements that these portions of text must not contain, named *exclusion zones*.

A Contextual Rule has this form:

$$Label \rightarrow LeftContext \backslash Body \,/ RightContext;\ SetsSpecification$$

where:

- *LeftContext*, *RightContext* and *Body* are sequences of two types of elements: labels to be recognized or exclusion zones.
- *LeftContext* and *RightContext* are the context within *Body* must occur.
- *Label* is the label assigned to *Body* if the sequence is detected.
- An *Exclusion Zone* is an expression with form **(ExcludedSet, Size)*, where *ExcludedSet* is the name of a set of labels and *Size* a natural number.
- *SetsSpecification* is the definition (by enumeration) of the sets mentioned in the exclusion zones of the rule. If there are no *Exclusion Zones*, *SetsSpecification* is not present in the rule.

For example, the following rule marks an initial modifier if a comma classified as CMMO is enclosing a segment of no more than 10 lexical units at the beginning of a sentence (marked by the label SENT).

$$modif \rightarrow \text{SENT} \setminus *(NoCm, 10) \,/\, \text{CMMO} \;;\; NoCm = \{\text{SENT}, \text{CM}\}$$

Given the text:

[.$^{\text{SENT}}$] Hasta hoy,$^{\text{CMMO}}$ pasados diez meses, no se sabe por qué mataron a Piazza, pero todos los indicios van en dirección a un crimen policial.

the application of this rule gives as result:

[.] $\boxed{\text{Hasta hoy}}^{modif}$, pasados diez meses, no se sabe por qué mataron a Piazza, pero todos los indicios van en dirección a un crimen policial.

The unit work for an analyzer is a module, a finite set of contextual rules. A module may contain several rules with the same left hand side (label). As usual in the generative grammar model, two rules with the same left hand side symbol are interpreted as disjunctive conditions for the deduction of this symbol. Modules, taken by a rule interpreter written in Prolog, are sequentially applied to the input text.

4.1 Rules

As mentioned at the beginning of this section, our purpose was to keep parsing rules as simple as possible, driving the analysis by punctuation and exploiting

the classifier information at maximum. Rules were written by hand, generalizing phenomena observed in our corpus.

The analysis is divided in 7 modules, each of them recognizes certain type of structure: simple series (33 rules); incises inside parenthesis (14 rules); incises delimited by parenthesis (2 rules); bipolar structures (3 rules); incises (26 rules); propositional series (16 rules); final analysis (12 rules).

In the following sections, we give some of the system rules with a brief natural language explanation.

Incises

– If the first comma occurs in the 20 first positions and it is valued as CMCO or CMCO, mark it as an initial incise.

$\text{incise}_{ini} \rightarrow$ SENT \ *(S,20) CMCO /; $S = \{$CM, SENT$\}$

$\text{incise}_{ini} \rightarrow$ SENT \ *(S,20) CMMO /; $S = \{$CM, SENT$\}$

– If a text of no more than 10 units with no finite verb is enclosed by a pair of commas labeled as CMIB and CMIE, label it as an incise.

incise $\rightarrow$ \ CMIB *(S,10) CMIE/; $S = \{$incise, CM, VERBFIN, SENT$\}$

Simple series

– If there is a comma labeled as CMSO, and near it a noun, a conjunction and another noun, but no verb, label the segment as part of a series ($series_{ac}$).

$\text{series}_{ac} \rightarrow$ CMSO \ $*\,(S, 5)$ NOM CONJ NOM/;
$S = \{$CONJ, CM, VERBFIN$\}$

– If there is a noun and a comma labeled as CMSO before a series, label it as a series.

$\text{series}_{ac} \rightarrow$ CMSO \ $*\,(S, 5)$ NOM CMSO series_{ac}/;
$S = \{$CONJ, CM, VERBFIN$\}$

Propositional segments

– If the last comma is labeled as CMSP and encloses a segment that begins with a conjunction and has a finite verb, label it as a propositional segment.

prop $\rightarrow$ CMSP \ CONJ *(S_1,50) VERBFIN*(S_2,50) / SENT;
$S_1 = \{$prop, conj, CMSP, SENT$\}$
$S_2 = \{$prop, CMSP, VERBFIN, SENT$\}$

– If a segment having a finite verb is between an initial incise and a CMSP comma, label it as a propositional segment.

prop $\rightarrow$ incise_{ini} \ *(S_1,20) VERBFIN *(S_2,20)/ CMSP;
$S_1 = \{$prop, incise_{ini}, CMSP, SENT$\}$
$S_2 = \{$prop, VERBFIN, CMSP, SENT$\}$

4.2 Results

For the analysis evaluation, we use 102 previously unseen sentences, with an average length of 25 words.

The following process is applied: first, text is tagged using Freeling; then, the classifier labels the commas; and, finally, the analyzer is applied. Thus, these results carry the errors of the three processes implied.

Table 3. Analysis results

	Precision	Recall	F-measure
Simple Series	75	38	**50**
Incises	95	77	**85**
Propositions	88	81	**84**

The analyzer correctly parsed 68 sentences, in which 90 propositions, 56 incises and 5 simple series were detected. Also, some structures were correctly labeled in the 34 remaining sentences: 15 propositions, 13 incises, and 1 simple series.

The F-measure for simple series is very low, 48%. Though, the errors are exclusively due to limits not correctly established (i.e. partial recognition): no segments where labeled as simple series erroneously. Errors in both tagger and comma-evaluator explain the low recall rate obtained.

In the other hand, propositions and incises have very good F-measure. In particular, incises are recognized with a precision of 95% with very simple rules. Both, the comma tagger and the proposed rules, correctly capture appositive incises, initial modifiers and discourse connectives. Recall is lowered by multi-valuated commas –not considered by the analyzer–, commas incorrectly labeled and complex structures not captured by the analyzer rules. The error rate for propositions segments is higher, because they are generally the "outer" structures of the sentence: their recognition are strongly influenced by the series and incises incorrectly labeled.

5 Conclusions

In this paper, we have presented a classification for commas in Spanish texts, based on a normative given by the Spanish Real Academy, but modified to cope with phenomena found in our corpus.

The classification is used in learning a comma value tagger using machine learning techniques. Although at a first glance the results are not excellent, the classifiers reduce the baseline error rate significantly. In fact, we achieved very good results if we considered that comma classification is (a) a difficult task even for humans, (b) it often depends on semantic factors, and (c) the classifiers only take POS tags and positional information as input attributes.

The best classifier is combined with a rewriting system for recognizing series, incises and propositional segments in texts. Despite the classifier error, very good

rates of precision and recall are obtained. It is remarkable that the interaction between two different approaches allowed us to write more simple rules to capture text phenomena, and precision for some structure recognition is very good.

We conclude that combining these two different approaches –machine learning techniques and a symbolic rule system– let us find a very good solution for our text structure task.

References

1. Murat Bayraktar, Bilge Say, and Varol Akman. An analysis of english punctuation: The special case of comma. *International Journal of Corpus Linguistics*, 3(1):33–57, January 1998.
2. Ted Briscoe and John Carroll. Developing and evaluating a probabilistic lr parser of part-of-speech and punctuation labels. In *Proceedings of the ACL/SIGPARSE 4th International Workshop on Parsing Technologies*, pages 48–58, Prague/Kqrlovy Vary, Czech Republic, 1995.
3. X. Carreras, I. Chao, L. Padró, and M. Padró. Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, June 2004.
4. Bernard Jones. Can punctuation help parsing? Technical Report 29, Cambridge University, Cambridge, UK, 1994.
5. J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, US, 1993.
6. Real Academia Española. *Diccionario Panhispánico de Dudas*. Santillana, Bogotá, Colombia, 2005.
7. Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.
8. Sebastian van Delden and Fernándo Gómez. A finite state comma tagger. *International Journal on Artificial Intelligence Tools*, 13(3):449–468, 2004.
9. Michael White. Presenting punctuation. In *Proceedings of the Fifth European Workshop on Natural Language Generation*, pages 107–125, Leiden, the Netherlands, May 1995. Faculty of Social and Behavioural Sciences, University of Leiden.
10. Dina Wonsever and Jean-Luc Minel. Contextual rules for text analysis. *Lecture Notes in Computer Science*, 2004:509–521, 2001.

Unifying Nondeterministic and Probabilistic Planning Through Imprecise Markov Decision Processes*

Felipe W. Trevizan[1,**], Fábio G. Cozman[2], and Leliane N. de Barros[1]

[1] Instituto de Matemática e Estatística, Universidade de São Paulo
Rua do Matão, 1010, Cidade Universitária – 05508-090 São Paulo, SP, Brazil
{trevisan, leliane}@ime.usp.br
[2] Escola Politécnica, Universidade de São Paulo
Av. Prof. Mello Moraes, 2231, Cidade Universitária – 05508-900 São Paulo, SP, Brazil
fgcozman@usp.br

Abstract. This paper proposes an unifying formulation for nondeterministic and probabilistic planning. These two strands of AI planning have followed different strategies: while nondeterministic planning usually looks for minimax (or worst-case) policies, probabilistic planning attempts to maximize expected reward. In this paper we show that both problems are special cases of a more general approach, and we demonstrate that the resulting structures are Markov Decision Processes with Imprecise Probabilities (MDPIPs). We also show how existing algorithms for MDPIPs can be adapted to planning under uncertainty.

1 Introduction

Planning is not only ubiquitous in artificial intelligence; it also appears in many different forms. While classical planning focuses on deterministic settings without any uncertainty, several non-classical approaches have tried to deal with various forms of uncertainty [1]. Among these approaches, *probabilistic planning* has produced significant results in recent years [2,3,4]. Another important approach is *nondeterministic planning* [5], where one does not even assign probabilities to the consequences of actions.

A particularly apt perspective from which to read this literature is due to Geffner and Bonet [6]. The idea is to capture what is common across approaches by formulating general languages, models, and algorithms. As discussed in Section 2, this perspective has been quite effective in unifying various strands of planning, from classical to probabilistic, including variants of nondeterministic planning. A unified understanding of planning problems is obviously beneficial not only to artificial intelligence but to several other fields such as operations research and management.

* Project funded by Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP) process number 04/09568-0.
** Funded by Conselho Nacional de Pesquisa (CNPq).

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 502–511, 2006.

The just mentioned general formulation takes probabilistic and nondeterministic approaches as two extreme and unrelated positions concerning planning. They are based on different assumptions concerning beliefs (either always translated into probabilities, or never translated into probabilities), and different prescriptions for action (either focused on average behavior through expected utility, or on worst-case guarantees coming from minimax). Accordingly, communities in probabilistic and nondeterministic planning have had little real interaction. In a sense, this is the general decision-theoretic contrast between Bayesian position that prescribes expected utility, and a minimax position that looks at worst case behavior. But in decision theory there are many other options, and in particular there are interesting options that can handle not only expected and minimax positions, but also other positions in between. Thus one can have a decision problem where some events have probability values attached to them, while other events may be associated with "nondeterministic" phenomena.

In this paper we propose a unifying formulation for planning problems, where we can smoothly transition between probabilistic and nondeterministic planning. These two approaches are viewed as simple special cases, and our analysis reveals a spectrum of new planning problems that has not been considered by the literature in artificial intelligence so far. We demonstrate that the resulting structures are Markov Decision Processes with Imprecise Probabilities (MDPIPs), a model proposed in operations research to solve control problems. We also show how existing algorithms for MDPIPs can be adapted to planning under uncertainty.

The remainder of this paper is organized as follows. In Section 2 we summarize Geffner and Bonet's unifying perspective on planning — thus defining the probabilistic and nondeterministic varieties. Section 3 introduces basic concepts underlying risk and uncertainty. Section 4 defines our proposal model for planning under uncertainty (PUU), named **PUU model**. In Section 5 we demonstrate that the PUU model is a variant of Markov Decision Processes with Imprecise Probabilities (called **MDPIPs** in the literature). Section 6 adapts MDPIP algorithms for PUU models. Finally, in Section 7 we draw some conclusions.

2 Planning Models

We briefly review the mathematical models needed to characterize planning tasks with full observability for different action dynamics (partial observability can be addressed with minor changes in the framework). Every state model that we consider can be defined in terms of the following basic state model [6]:

BSM1 a discrete and finite state space $\mathcal{S}$,
BSM2 a non-empty set of initial states $S_0 \subseteq \mathcal{S}$,
BSM3 a goal given by a non-empty set $S_G \subseteq \mathcal{S}$,
BSM4 a non-empty set of actions $\mathcal{A}(s) \subseteq \mathcal{A}$ representing the actions applicable in each state s,
BSM5 a state transition function $F(s, a) \subseteq \mathcal{S}$ mapping states s and actions $a \in \mathcal{A}(s)$ into non-empty sets of states, i.e. $\|F(s, a)\| \geq 1$, and
BSM6 a positive action cost $C(a, s)$ for doing $a \in \mathcal{A}(s)$ in s.

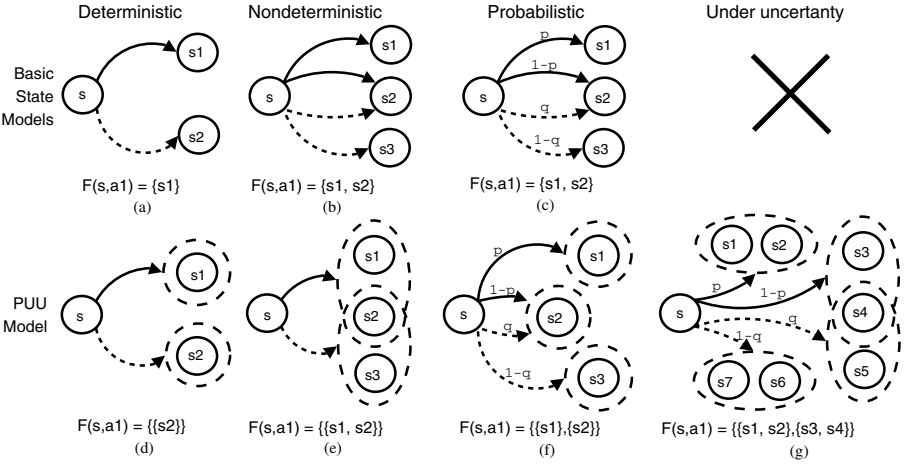


Fig. 1. This figure illustrates the structure of transition function in the different models discussed in this paper. Solid and dotted arcs represents different actions. Solid circles are states and dashed circles indicates *possible-state sets*. Note that under uncertainty scenarios cannot be represented by the basic state model.

Differents models can be defined adding new restrictions or modifing the statements 2, 5 and 6. Those models, depicted by the first row of Fig. 1, are:

- **Deterministic Models** (Fig. 1 (a)), where the dynamics are defined by a deterministic state transition function, i.e., $\|F(s,a)\| = 1$. This is the basis of the *classical* planning scenario, where one has additional constraints of initial state $\|S_0\| = 1$ and $\mathcal{C}(a,s) = 1 \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$. A sequence of actions $a_0, \ldots, a_{n-1}$, called plan, is a valid solution to the model if for $0 \leq i \leq n-1 \quad s_{i+1} \in \mathcal{F}(s_i, a_i)$, $a_i \in \mathcal{A}(s_i)$ and $s_n \in F(s_{n-1}, a_{n-1}) \cap S_G$.
- **Nondeterministic Models** (Fig. 1 (b)), where the actions may result in more than one successor state without preferences among them. So we have the same model as in deterministic planning, but uncertainty in actions. In fact, the term "nondeterminism" should here be understood as "automata-style" nondeterminism and using the terminology discussed in Section 3, we actually have *planning under pure Knightian uncertainty*. Since we assume full observability, a valid solution to the model is a policy, i.e. a function $\pi : \mathcal{S} \to \cup_{s \in \mathcal{S}} \mathcal{A}(s)$, that is *closed and proper* with respect to S_0 [6]. In this model, a policy offers guarantees about the worst-case behavior of the environment.
- **Probabilistic Models** (Fig. 1 (c)), where actions have probabilistic consequences. Not only the function $\|F(s,a)\| \geq 1$ is given, but also the model includes a probability distribution $P(\cdot|s,a)$ over $F(s,a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$. As in the Nondeterministic Models, a solution to the model is a policy, but in this case the objective is to maximize expected behavior — where expected behavior is quantified through a *single* probability measure.

There are algorithms that compute policies for each one of these problems. A recent development is the derivation of a single algorithm that can be instantiated for different models, including the ones just described [2]. However it should be emphasized that this generalized formulation does not yield a smooth family of solutions that moves from one case to the other. In particular, there is no algorithm that has the probabilistic and nondeterministic cases as special ones, and also that copes with mixtures of these cases. The main goal of this paper is to start the construction of such a framework.

3 Risk, Knightian Uncertainty and Sets of Probabilities

Instead of moving directly to our general formulation, it is instructive to start with an open-minded review of decision theory. Here a decision maker contemplates a set of options (in our setting, policies); each option yields a utility depending on the state of nature that obtains [7]. We consider a set of states of nature Ω; then each option is a function $f : \Omega \to \Re$. If a decision maker can specify a single probability measure P over a field of events defined on Ω, then this "Bayesian agent" will evaluate each option f by expected utility, $E_P[f]$. Typically one assumes that such an agent can select any option that is dominated by expected utility — a simple criterion that leads to a rich theory [8].

However, there may be situations where an agent does not have a single probability measure. A common assumption then is that the agent will have *no* probability at all. The usual solution then is to look at worst-case scenarios: select f that displays the highest worst utility — a minimax solution [7]. The difference between these extremes (one/no probability) is well studied in economics and psychology. Usually the presence of probabilities is associated with the expression *risk*, while the absence of probabilities is associated with *uncertainty*, or rather, *Knightian uncertainty* (from the work of Knight [9]). To indicate the pervasiveness of these concepts in economics practice, it suffices to quote from a relevant speech by Alan Greenspan, read in January 3 2004:

> ...uncertainty is not just a pervasive feature of the monetary policy landscape; it is the defining characteristic of that landscape. The term "uncertainty" is meant here to encompass both "Knightian uncertainty," in which the probability distribution of outcomes is unknown, and "risk," in which uncertainty of outcomes is delimited by a known probability distribution...

Now it is clear that *sequential decision making under risk* is *probabilistic planning*, while *sequential decision making under Knightian uncertainty* is *nondeterministic planning*. In fact, we would like to suggest that the term "nondeterministic" is an unfortunate one in the present setting, as nondeterminism usually suggests some form of probabilistic model. It seems that Knightian uncertainty, although longer, is a less overloaded term.

Once it is recognized that risk and Knightian uncertainty are two challenges a decision maker may face, one is naturally lead to ask about situations of both risk *and* Knightian uncertainty. That is, we may consider the possibility that an

agent displays imprecision in probability values or even that the agent considers a set of probability values. There are many reasons where such a general situation may arise. First, it may happen that existing beliefs are incomplete or vague [10,11,12], either because there is no resources to spend in their elicitation, or because experts are psychologically unable to specify precise probability values. Second, it may be the case that a group of experts disagrees on probability values, and no compromise can be reached other than the collection of their opinions [13,14]. Another reason to abandon a single probability measure is when one is interested in the robustness of inferences — that is, in evaluating how much inferences can change when probability values are allowed to vary [8,15,16].

Our strategy in this paper is, at a fundamental level, simple: we intend to bring the decision theory of risk *and* Knightian uncertainty to the realm of artificial intelligence planning. In this setting, uncertainty will be represented by *sets of probability measures*. At one extreme, we obtain probabilistic planning (all sets are singletons); at the other extreme, we obtain nondeterministic planning (all sets are as large as possible). Moreover, we obtain a continuum of models as we allow sets of probability measures to transit from vacuously large ones to singletons.

Artificial intelligence has witnessed steady interest in sets of probability measures, for example, in the theory of probabilistic logic [17,18,19], in Dempster-Shafer theory [20], in theories of argumentation [21], and in techniques that generalize graph-theoretic models such as Bayesian networks [22,23,24].[1] Our contribution here is to identify the planning under uncertainty spectrum with the theory of sets of probability measures.

4 Planning Under Uncertainty: The Risk and Knightian Uncertainty Spectrum

The *planning under uncertainty model*, referred to here as the **PUU model**, is a more general model as it gives a precise semantics to planning tasks involving nondeterministic and probabilistic *effects* of actions. A planning problem can be solved considering, simultaneously, these two types of action's effects.

Since the PUU model has to represent nondeterministic effects, the transition function $F(s, a)$ (from the basic state model described in Section 2) must be understood as follows. Instead of taking $F(s, a) \subseteq \mathcal{S}$ as before, now we map states and actions to sets of sets of the state space. That is, for all k in $F(s, a)$, k is a subset of or equal to $\mathcal{S}$.

Definition 1. *In the PUU model, the transition function $F(s, a)$ maps states s and actions $a \in \mathcal{A}(s)$ into nonempty sets of the parts of $\mathcal{S}$: $F(s, a) \subseteq 2^{\mathcal{S}}$.*

Definition 2. *A **possible-state set** k is a set composed of possible resulting states achieved with the execution of an action a; that is, $k \in F(s, a)$ with $F(s, a)$ the state transition of Definition 1.*

[1] There is now significant literature on the theory and applications of sets of probability measures [25,26,27,28].

With the above definitions the probabilistic function $P(k|s,a),\ k \in F(a,s)$ has a novel interpretation in the PUU model: $P_n(k|s,a)$ represents the probability of the next state to be within one of the states in k. The PUU model is illustrated in the second row of Fig. 1 and a formal description of this model is given by:

PUU1 a discrete and finite state space $\mathcal{S}$,
PUU2 a nonempty set of initial states $S_0 \subseteq \mathcal{S}$,
PUU3 a goal situtation given by a nonempty set $S_G \subseteq \mathcal{S}$,
PUU4 a nonempty set of actions $\mathcal{A}(s) \subseteq \mathcal{A}$ representing the actions applicable in each state s,
PUU5 a state transition function $F(s,a) \subseteq 2^{\mathcal{S}}$ mapping states s and actions $a \in \mathcal{A}(s)$ into nonempty sets of the parts of the state space,
PUU6 a probability distribution $P_n(\cdot|s,a)$ over $F(s,a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$ where $P_n(k|s,a)$ represents the probability of choosing the possible-state set $k \subseteq \mathcal{S}$ when action a is applied in state s, and
PUU7 a positive action cost $C(a,s)$ for doing $a \in \mathcal{A}(s)$ in s.

Notice that there are two types of choices happening the PUU model: a probabilistic choice of a possible-state set and a nondeterministic choice of a sucessor state from the possible-state set. The planning under uncertainty task can thus be characterized by domains for which the action dynamics satisfies the following restrictions: (1) $\|F(s,a)\| > 1$ and (2) $\exists k \in F(s,a) \text{s.t.} \|k\| > 1$, for $s \in \mathcal{S}, a \in \mathcal{A}(s)$ (Fig. 1 (d)). If none of these requirements is true, then PUU model is reduced to one of the models described in Section 2.

If the first requirement is false, i.e. $\|F(s,a)\| = 1$, and the second is true, the PUU model is equivalent to the nondeterministic model (Fig. 1 (e)). This is because: $\forall s \in \mathcal{S}\ ,\ a \in \mathcal{A}(s)$, if $\quad \|F(s,a)\| = 1$ then $P_n(k \in F(s,a)|s,a) = 1$, which means that the choice of a possible-state set will be deterministic while the occurrence of a single state $s' \in k$ will be nondeterministic.

For the planning set where the first requirement is true and the second is false, then the model corresponds to the Probabilistic Model (Fig. 1 (f)). This is due to the fact that $\forall s \in \mathcal{S}, a \in \mathcal{A}(s), k \in F(s,a) \quad \|k\| = 1$, implying that there will be only one candidate to the nondeterministic choice, with probability $P_n(k|s,a)$ after executing a in the state s. Under this assumptions the probability distribuition over $2^{\mathcal{S}}$ is equivalent to a probability distribuition over $\mathcal{S}$.

Finally, when both requirements are false, the model is equivalent to the deterministic model (Fig. 1 (d)) once there is no point of choice: neither in the probabilistic choice of a possible-state set nor in the nondeterministic choice of a sucessor state.

Furthermore, the complete PUU model is equivalent to a Markov Decision Process having imprecisely known transition probabilities. This equivalence, proved in the next section, gives a formal semantics for the PUU model.

5 The Relation Between PUU and MDPIP Model

Markov Decision Processes with Imprecise Probabilities (MDPIPs) [29,30] are an extension of Markov Decision Processes (MDPs) [31] where the probabilities

describing the transition between states are not defined as a number, but as a finite set of linear inequalities. Consequently, the possible effects of an action are modelled by a credal set $\mathcal{K}$ [23] over the state space instead of a probability distribution over the same space. A precise definition of an MDPIP is:

MIP1 a discrete and finite state space $\mathcal{S}$,
MIP2 a goal situtation given by a nonempty set $S_G \subseteq \mathcal{S}$,
MIP3 a nonempty set of actions $\mathcal{A}(s) \subseteq \mathcal{A}$ representing the actions applicable in each state s,
MIP4 a nonempty credal set $\mathcal{K}_s(a)$ representing the possibles probability distributions $P(\cdot|a, s)$ over $\mathcal{S}$, and
MIP5 a positive action cost $C(a, s)$ for doing $a \in \mathcal{A}(s)$ in s.

The formulation above is based on Game Theory and considers the existence of a mechanism that selects the exact probability distribution after an action has been selected. This mechanism is usually called nature and an MDPIP can be solved only if an assumption is made about its behavior. In this paper, we assume that nature is intent on maximizing the expected total discount cost for each state that the plannet wishes to minimize (1). Therefore, a minimax criterion is adopted to find a policy.

$$V(s) = \min_{a \in \mathcal{A}(s)} \max_{P(\cdot|s,a) \in \mathcal{K}_s(a)} \{C(a, s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a)V(s')\} \tag{1}$$

In [30] it has been shown that the solution to (1), called $V^*(s)$, exists and is unique. It is also proved that the optiomal policy for an MDPIP can be expressed by a stationary policy, i.e., the same policy for any instant in time. We have the following fundamental relationship between the PUU model and MDPIPs:

Proposition 1. *The PUU model is a special case of the MDPIPs model.*

Proof. Note that PUU1, PUU3, PUU4 and PUU7 are equal, respectively, to MIP1, MIP2, MIP3 and MIP5. Thus the proof is reduced to prove that PUU5 and PUU6 implies in MIP4.

First, note that PUU6 bounds the probability of being in state s' after applying the action a in the state s by (2). This is due to the definition of possible-state set: let $k \in F(s, a)$, if $s' \notin k$, then nature is not able to choose s' as a non-deterministic effect of a.

$$P_n(\{s'\}|s, a) \leq P(s'|s, a) \leq \sum_{\substack{k \in F(s,a) \\ s' \in k}} P_n(k|s, a) \leq 1 \quad \forall s' \in \mathcal{S} \tag{2}$$

Let us define the set of states $\mathcal{D}(k, s, a)$, for (3) . This set represents all non-deterministic effects of k that bellong only to k.

$$\mathcal{D}(k, s, a) = k \setminus \bigcup_{\substack{k' \in F(s,a) \\ k' \neq k}} k' \tag{3}$$

From PUU5 and PUU6 it is possible to bound the sum of the probabilities of each state in a possible-state set $k \in F(s,a)$ and in the associated set $\mathcal{D}(k,s,a)$. These bounds are presented in (4).

$$0 \leq \sum_{s' \in \mathcal{D}(k,s,a)} P(s'|s,a) \leq P_n(k|s,a) \leq \sum_{s' \in k} P(s'|s,a) \leq 1 \tag{4}$$

The set of inequations (2) and (4) for a state $s \in \mathcal{S}$ and an action $a \in \mathcal{A}(s)$ describe a possible credal set $\mathcal{K}_s(a)$ for MIP4. □

Proposition 1 not only makes the results in [29,30] valid for the PUU model, but also suggests algorithms to solve it. An algorithm that can solve a PUU problem, inspired by previous algorithms for MDPIPs, is given in the next section.

6 Algorithms

Due to Proposition 1, every algorithm for MDPIPs can be applied to solve a PUU problem. However, the process is not immediate as it is necessary to adapt several key concepts. To illustrate this, we selected a modified version of the policy-iteration algorithm [31] given by [30]. This algorithm is divided in two phases, *policy evaluation*, where the expected utility of a policy is calculated, and *policy improvement* in which a better policy is built based on the values received from the *policy evaluation* phase.

The algorithm presented bellow receives an tuple $\langle \mathcal{S}, \mathcal{A}, K_s(a)$ (constructed by proposition 1) $, C(a,s)\rangle$, a discount factor γ and an initial, possible random, policy π as input and return the optimal policy for this PUU problem.

Policy evaluation. For every state $s \in \mathcal{S}$:

(a) Select a probability distribution $P(\cdot|s,\pi(s)) \in \mathcal{K}_s(\pi(s))$.
(b) Use $P(\cdot|s,\pi(s))$ to solve (5).

$$V_\pi(s) = C(\pi(s),s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V_\pi(s') \tag{5}$$

(c) Find the probability distribution $P'(\cdot|s,\pi(s)) \in \mathcal{K}_s(\pi(s))$ s.t.

$$\max_{P'(\cdot|s,\pi(s)) \in \mathcal{K}_s(\pi(s))} \{C(\pi(s),s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,\pi(s)) V_\pi(s'), \tag{6}$$

where $V_\pi(s')$ are solutions to (5).

(d) If $V_\pi(s)$ is equal to $C(\pi(s),s) + \gamma \sum_{s' \in \mathcal{S}} P'(s'|s,a) V_\pi(s')$ then proceed to the *policy improvement* phase, otherwise, return to step (b) with $P(\cdot|s,\pi(s)) \leftarrow P'(\cdot|s,\pi(s))$.

Policy improvement. For every state $s \in \mathcal{S}$, find $\pi'(s)$ s.t.

$$\pi'(s) = \operatorname*{argmin}_{a \in \mathcal{A}(s)} \max_{P(\cdot|s,a) \in \mathcal{K}_s(a)} \{C(a,s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V_\pi(s')\}, \tag{7}$$

using the value $V_\pi(s')$ from (5). If $\pi'(s)$ is different from $\pi(s)$ for some state $s \in \mathcal{S}$, return to the *policy evaluation* phase with $\pi \leftarrow \pi'$; otherwise, π is the minimax optimal solution for the PUU problem.

7 Conclusions and Future Work

We can list the following contributions of this work. First, we identified a general formulation that encompasses probabilistic and nondeterministic planning, *and that includes* a continuum of planning problems between these extremes. In fact, the probabilistic/nondeterministic actions we define open new types of planning scenarios that go beyond existing planning problems. For example, one may have actions whose transitions are specified by general sets of probabilities, not just the set-valued consequences discussed in this paper. Second, we have shown how our proposal fits within the MDPIP framework. Proposition 1 establishes the link between planning under risk and Knightian uncertainty and the previous literature on MDPIPs. We have then shown how to transfer algorithms previously developed for MDPIP to our proposal.

This paper should open a profitable avenue for future research in a variety of directions. It would be interesting to explore the many possible combinations of probabilistic and nondeterministic actions, and more general transitions defined by sets of probability measures. However, we feel that the most fruitful task for the near future is to adapt existing online algorithms for existing planning problems, such as RTDP and LRTDP [3], to the full generality of planning under risk and Knightian uncertainty.

References

1. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory & Practice. Morgan Kaufman (2004)
2. Bonet, B., Geffner, H.: Learning Depth-First Search: A unified approach to heuristic search in deterministic and non-deterministic settings, and its application to MDPs. In: ICAPS, To Appear (2006)
3. Bonet, B., Geffner, H.: Labeled RTDP: Improving the convergence of real-time dynamic programming. In: ICAPS, Trento, Italy, AAAI Press (2003) 12–21
4. Guestrin, C., Koller, D., Parr, R., Venkataraman, S.: Efficient solution algorithms for factored MDPs. J. Artif. Intell. Res. (JAIR) **19** (2003) 399–468
5. Bertoli, P., Cimatti, A., Roveri, M., Traverso, P.: Planning in nondeterministic domains under partial observability via symbolic model checking. In: IJCAI. (2001) 473–478
6. Bonet, B., Geffner, H.: Planning with incomplete information as heuristic search in belief space. In: ICAPS, Breckenridge, CO, AAAI Press (2000) 52–61
7. Luce, D., Raiffa, H.: Games and Decisions. Dover edition, Mineola (1957)
8. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer-Verlag (1985)
9. Knight, F.H.: Risk, Uncertainty, and Profit. Hart, Schaffner & Marx; Houghton Mifflin Company, Boston (1921)

10. Levi, I.: The Enterprise of Knowledge. MIT Press (1980)
11. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
12. Walley, P.: Measures of uncertainty in expert systems. AI **83** (1996) 1–58
13. Seidenfeld, T., Kadane, J.B., Schervish, M.J.: On the shared preferences of two Bayesian decision makers. The Journal of Philosophy **86**(5) (1989) 225–244
14. Seidenfeld, T., Schervish, M.: Two perspectives on consensus for (Bayesian) inference and decisions. IEEE Transactions on Systems, Man and Cybernetics **20**(1) (1990) 318–325
15. Huber, P.J.: Robust Statistics. Wiley, New York (1980)
16. Kadane, J.B., ed.: Robustness of Bayesian Analyses. Volume 4 of Studies in Bayesian econometrics. Elsevier Science Pub. Co., New York (1984)
17. Frisch, A.M., Haddawy, P.: Anytime deduction for probabilistic logic. Artificial Intelligence **69** (1994) 93–122
18. Halpern, J.Y.: Reasoning about uncertainty. MIT Press (2003)
19. Nilsson, N.J.: Probabilistic logic. Artificial Intelligence **28** (1986) 71–87
20. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
21. Anrig, B., Bissig, R., Haenni, R., Kohlas, J., Lehmann, N.: Probabilistic argumentation systems: Introduction to assumption-based modeling with ABEL. Technical Report 99-1, Institute of Informatics, University of Fribourg (1999)
22. Cozman, F.G.: Credal networks. AI **120** (2000) 199–233
23. Cozman, F.G.: Graphical models for imprecise probabilities. International Journal of Approximate Reasoning **39**(2-3) (2005) 167–184
24. Fagiuoli, E., Zaffalon, M.: 2U: An exact interval propagation algorithm for polytrees with binary variables. Artificial Intelligence **106**(1) (1998) 77–107
25. de Cooman, G., Cozman, F., Moral, S., Walley, P., eds.: Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications (SIPTA). Universiteit Gent, Ghent, Belgium (1999)
26. de Cooman, G., Fine, T.L., Seidenfeld, T.: Proceedings of the 2nd International SIPTA Shaker Publishing, The Netherlands (2001)
27. Bernard, J.M., Seidenfeld, T., Zaffalon, M., eds.: Proceedings of the 3rd International SIPTA Carleton Scientific, Lugano, Switzerland (2003)
28. Cozman, F.G., Nau, R., Seidenfeld, T.: Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications. SIPTA (2005)
29. White III, C.C., Eldeib, H.K.: Markov decision processes with imprecise transition probabilities. Operations Research **42**(4) (1994) 739–749
30. Satia, J.K., Jr, R.E.L.: Markovian decision processes with uncertain transition probabilities. Operations Research **21**(3) (1973) 728–740
31. Howard, R.A.: Dynamic Porgramming and Markov Processes. MIT Press (1960)

Achieving Conditional Plans Through the Use of Classical Planning Algorithms

Jony Teixeira de Melo and Carlos Roberto Lopes

Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
Av. João Naves de Ávila 2160 – 38400-902 –Uberlândia/MG, Brazil
jony@pos.facom.ufu.br, crlopes@facom.ufu.br

Abstract. In this work we concentrate on generating plans that take into account conditional actions. The main idea was to develop an algorithm that extended classical formalisms in a general way. By general way, we mean a flexible formalism that could use any algorithm for classical planning without any change. The result of our efforts was the development of a planner that we called *METAPlan*. In this paper we describe *METAPlan* and show some results of its performance.

1 Introduction

Most classical planners use the assumption that there is no uncertainty in the world: every action has a predictable output. However, uncertainties are rules rather than exceptions in a real world environment. Therefore, we need a planner that could have different outputs depending on the action that is required to fulfill a goal. That means that the majority of the classical planners cannot be used for this purpose.

Several approaches for planning were developed to deal with uncertainties [1]. Contingent Planning is one of these approaches. A contingent plan is a plan that contains actions that may or may not actually be executed, depending on the circumstances that hold at the time. A contingent planner must be able to produce plans even if the initial conditions and the outcomes of some of the actions are not known. Several contingent planners were developed. Cassandra[2] is a partial-order, contingency, domain-independent problem solver architecture based on UCPOP [3]. SGP[4] is an extension of the Planning Graph Analysis algorithm GRAPHPLAN[5] to support uncertainty in initial conditions and actions. CNLP[6] uses the basic SNLP algorithm to construct contingents plans, PLINTH[7] is a total order plan very similar to CNLP in its treatment of contingency plans. SENSp[8] like Cassandra is based on UCPOP but differs in the way it represents uncertainty. All have in common the use of extended STRIPS-based action representation[9].

In this work we developed METAPlan, which is a planner that takes into account conditional effects. METAPlan is generic in the sense that it can make use of any classical planning algorithm. Therefore, it may use UCPOP, GRAPHPLAN, FF or any other algorithm. The terminology we use for describing actions is based on modeling of processes existing in the workflow literature. By using this terminology is possible to represent concurrency, sequencing, and conditional effects of actions. We introduced this terminology in section 2. We could have used other planning

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 512 – 521, 2006.

language for representing actions, such as PDDL[10]. Our choice was based on the fact that we are investigating the use of planning in the modeling and enactment of workflows. In order to make use of classical algorithms, it is necessary to decompose actions with conditional effects into actions with simple effects. As a result of this decomposition several plans can be created. The next step in this process is to merge these plans into a unique plan with conditional effects. The algorithms for decomposing actions and generating plans are described in section 3. In section 4, we describe results that we achieved with METAPlan. Related work is presented in section 5. In section 6, we conclude and present some lines of further research.

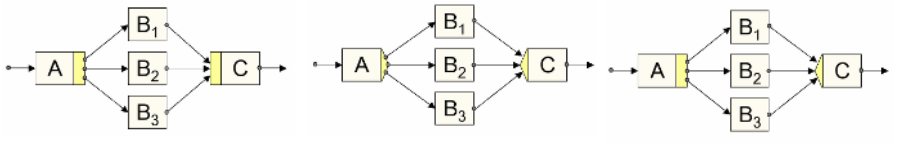
2 Modeling Activities in Workflow Systems

According WfMC's[1] definition, workflow is the total or partial automation of business processes that makes it possible to pass documents, information or activities from one participant to another, according to a set of rules [11].

In the workflow literature the term "activity" is related to actions in the planning field. Below we describe the concept of activity and how activities can be chained.

Activities. The main element in a workflow is an activity that must be executed for the fulfillment of a goal. To execute a particular activity an actor must satisfy a set of attributes related to this activity. An actor can be a person, an application, or a computerized agent.

Chaining of Activities. The activities in a workflow can be organized in three different ways: *sequentially*, in *parallel* or *conditionally*. Sequentially means that as soon as an activity is executed, the following is activated. The subsequent activity cannot be initiated until the current one is concluded. Activities executing in parallel are activated simultaneously. They do not need to be concluded at the same time, because they can follow different criteria or they can demand distinct operations. As time passes, these parallel flows converge and become a sequential flow or else the end of the process. The conditional chaining happens when the next activity to be executed is based on a decision.



a) Parallel **b) Conditional** **c) Parallel with End Selection**

Fig. 1. Model of Branching Activities

Figure 1 depicts the three combinations of activities. A ***parallel*** flow allows the simultaneous execution of two or more activities from an *AND-Split* activity. Later, the flow will converge to an *AND-Join* connection. A ***conditional*** flow implies that

[1] *Workflow Management Coalition Specification*: An organization created in 1993 by about 90 companies (among them: HP, IBM, Microsoft, Oracle, SAP, Siemens and Xerox) whose objective is the development of standards for workflow technologies.

only one outcome will be enabled. Later the flow will converge to an *OR-Join* connection, which means that it needs to be reached by only one outcome. Finally, the flow ***parallel with end selection*** allows the simultaneous execution of two or more activities. However, the convergence point must be reached by only one activity.

3 Description of METAPlan

In this section we show how to generate a plan that contains conditional actions. Firstly, activities need to be changed in order to present only a single output. Secondly, the new activities, which result from the early step, are taken to a classical planner. As a result from this second step, several plans are achieved. These plans must be merged in order to form a unique plan containing the original activities which satisfy the goal.

3.1 Classical Planning

The work developed is based on the classical model of planning. In this way, planning is defined by a tuple of three elements ($\mathcal{A}$, I, $\mathcal{G}$), in which $\mathcal{A}$ describes the set of actions, I represents the initial state, and $\mathcal{G}$ corresponds to the goal to be achieved. Let $\mathcal{P}$ be the set of all propositions that represents facts in the world. The current state, or world, is assigned to w and represents the subset of satisfied propositions in $\mathcal{P}$ so that $w \subseteq \mathcal{P}$ in the world. In STRIPS, an action is represented by a triple (*pre(a), add(a), del(a)*) whose elements belong to the set of propositions $\mathcal{P}$ and corresponds respectively to its preconditions and effects – this last through the add and delete lists. Action a is applicable in w if $w \supseteq pre(a)$ holds. To apply a in w, replaces w with w' so that $w' = w - del(a) + add(a)$. It is assumed that $del(a) \cap add(a) = \{\}$. An action sequence is a plan if the result of its execution achieves its goal.

3.2 Mapping Activities in Classical Planning

There is a problem when mapping activities into operators following the classical planning formalism. Due to the possibility of flow splitting, direct mapping is not possible. Take the process of credit approval. Once a credit request is completed, an agent should execute activity *A* to determine whether the client has credit or not. Using direct mapping, it would be impossible for a classical planner to find a plan. This happens because the effects of this action (*Checking the credit*) are inconsistent and the result may be either credit approval or credit rejection.

In order to overcome this obstacle, we propose the decomposition of activity *A* into subactivities, one for each possible independent effect. This decomposition is valid only at planning time. Figure 2b shows the decomposition of activity *A* into two subactivities, one for each disjunctive effect (*P and* $\neg P$). Subactivity A'_1 allows the enabling of activity *B*; and subactivity A'_2 makes possible the execution of activity *C*. In this situation, *A* effects are replaced by a list of effects that make it possible to establish a causal link between *A* and *A'* (In our example the causal link is established by the clauses Q_{a1} and Q_{a2}). In each subactivity, the disjunctive effect of the original

activity and other terms that correspond to the negation of the causal link appear. This is done so that the planner does not take one of the subactivities into account without considering the main activity.

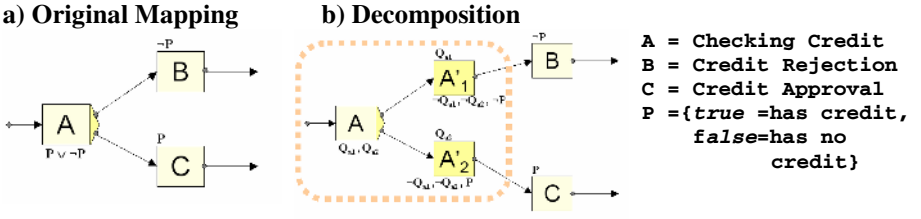


Fig. 2. Decomposing activities OR-Split

As an example consider the diagram in Figure 2 and suppose that the diagram represents a segment of a process that corresponds to a loan request from a client. The client can provide either a warrantor or a property as a guarantee for a loan. In this way, the activities in Figure 2 have the following meaning: *A = Accept a loan request; B = Loans with warrantor guarantee; C = Loans with property guarantee;*

```
Decompose_Activities (N){
01: A = elements_in(N);
02: R = {};
03: for i=1 in length(N) do
04:    if is_type_ORSplit(A[i])
05:    then
06:       T = extract_disj_term(A[i].effects);
07:       for k=1 in length(T) do
08:         Q   = new_term();
09:         LC  = LC  ∪ {Q};
10:         nLC = nLC ∪ {~Q};
11:       end for;
12:       for k=1 in length(T) do
13:         A' = new_activity();
14:         A'.precond = LC[k];
15:         A'.effects = nLC ∪ T[k];
16:         R = R ∪ {A'};
17:       end for;
18:       Tc=extract_conj_term(A[i].effects);
19:       A[i].effects = Tc ∪ LC;
20:    end if;
21:    R = R ∪ {A[i]};
22: end for
23: return R }
```

Fig. 3. Algorithm: Decomposition of activities

The algorithm for activity decomposition is described in Figure 3. The input data for the algorithm correspond to a set of activities *(N)*. The algorithm returns a set *(R)* with subactivities together with the original and modified *OR-Split* and *OR-Join* connections.

For each activity A_i we have to check whether or not it is an *OR-Split* activity. This is done by checking the existence of disjunctive effects in its description. Granted that we have an *OR-Split* activity, the next step consists in creating two new sets of terms, *Lc* and *nLc*. The first contains terms that represent causal links between the activity and the subactivities that result from its decomposition. The second contains terms that correspond to the negation of the terms in *Lc* so that $\forall x \in LC \rightarrow \exists(\neg x) \in nLC$. These new terms in *Lc* and *nLc* have no semantic meaning. Again, they were created for establishing a causal link between an activity and its subactivities. After creating *Lc* and *nLc*, the algorithm processes disjunctive elements (that generate a subactivity *A'* for both *Lc* and *nLc*). The preconditions of each subactivity correspond to the terms created in the previous step, which establishes a causal link between the activity and its subactivity. The list of effects of the subactivity is composed of *nLc* and the disjunctive term that is being processed. Thus, it guarantees the maintenance of the original effects and the negation of terms in the causal link. This helps the planner not to use more than one subactivity. The new subactivity is included in the answer set. Finally, the set of effects of the decomposed activity is changed. Its new set of effects is composed of the union between the set of the original conjunctive terms and that of the causal link with the subactivities (*Lc*). The activity is included in the answer set after its modification.

After processing all the activities, the algorithm returns the answer set that includes original activities, subactivities and modified activities.

3.3 Plan Generation

Once the activities are modeled correctly and the *OR-activities* are decomposed, the next step consists in taking the activities to a planner. It is possible to choose any classical planner available. As a result of this work, we developed a tool called SisMAP that allows the choice of different algorithms.

A planner returns a plan, which is a sequence of actions from an initial state to a state that satisfies the goal. In the simplest case, modeling does not include *OR-activities*. Therefore, all the activities may be either sequential or have some degree of parallelism.

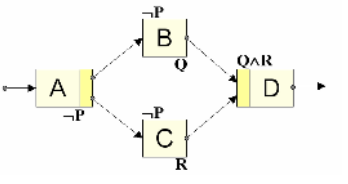
Definition 1. A *plan* $P = \{A_1,\ldots,A_n\}$ *is a set of activities so that* A_1 *is the first activity to be executed. The description* A_{i+1} *denotes the next activity to be executed where* i *is the index related to the current activity.*

Definition 2. Let $M=\{a_1,\ldots,a_k\}$ be a model of a process where each element a is a tuple given by (A_i, A_k) coming from a plan P so that $effects(A_i) \cap Pre(A_k) \neq \varnothing$ (not equal to the empty set), and $k > i$. This means that if there is a transition from one activity to another it is required that at least one of A_i effects be a precondition to A_k. A model of a process corresponds to a plan that can have actions with conditional effects and parallel actions.

Parallel Flow

Suppose that we have a model in which only sequential and parallel actions appear, that is, there are neither conditional activities nor parallel actions with final selection.

Figure 4 shows a workflow example with activities in parallel. A possible plan produced by the planner would be *P={A,B,C,D}*. However, this result does not inform ordering between each pair of activities. In this way, it is necessary to check whether each pair (A_i, A_{i+1}) satisfies **Definition 2** where $effects(A_i) \subseteq pre(A_{i+1})$. If this does not occur, the algorithm must look for an activity A_k that satisfies A_i so that $k>i+1$. When a pair is found, it is included in *M*. Then, the algorithm must look for an activity that comes before A_{i+1} in order to establish a link with it. For example, lets' consider that the diagram depicted in Figure 4 represents a segment of a process related to a *financial agent*. The financial agent can receive a loan request from a client. The client can provide a property as a guarantee. The business rules may establish a property evaluation and the checking of the client's financial status. Based on this consideration, we have the following meaning for the activities presented in Figure 4: *A = Loan request; B = Financial status checking; C = Property evaluation; D = Loan calculus based on financial status checking and property evaluation;*




```
P = {A,B,C,D}
M = {(A,B), (B,D), (A,C), (C,D)}
```

Fig. 4. A plan with activities in parallel

Given that there is no connection between *B* and *C* activities, their execution may occur simultaneously or not. Now, consider that one of the effects of *A* is a *request in wait*. In this way, *B* and *C* preconditions include *request in wait*. *B* effects consist of *financial checking concluded* while the *C* effect is *property evaluation finished*. Since *B* and *C* preconditions are the same, a planner could return either *{A,B,C,D}* or *{A,C,B,D}*. Suppose that the first solution is returned. According to the algorithm described in Figure 5, the first activity to be considered is A_i where $i=1$, thus $A_i=A$. In the next step, the pair (A_i, A_{i+1}) is checked with respect to ordering (using Definition 2). Given that *A* effects are equal to *B* preconditions, the pair *(A,B)* is included in model *M*.

Now consider activity A_i where $i=2$. In this case we have a situation where $A_i=B$ and pair (A_i,A_{i+1}), and where $A_{i+1}=C$, does not satisfy **Definition 2** (no element of the *B* effects is a member of *C* preconditions). Therefore, the pair *(B,C)* cannot be included in *M*. In order to handle this situation, the algorithm looks for activities following A_{i+1} until it finds one that makes up a pair with A_i that satisfies **Definition 2**. This processing is supported by a loop. In the first iteration, the pair *(B,D)* satisfies **Definition 2** (*B* effects are included in *D* preconditions). Consequently, *(B, D)* is included in *M* and the loop is finished. Next, the algorithm looks for an activity that establishes a pair with A_{i+1}. It looks for activities that come before A_{i+1}. In fact, if this does not happen the flowing parallel would not exist and the processing would not succeed. This search is accomplished by a new loop. The first iteration checks the pair (A_h, A_{i+1}) where $h=i-1$. With respect to the example, this pair corresponds to *(A, C)*. The pair *(A, C)* satisfies **Definition 2**. Therefore, it is included in *M* and the loop is finished.

```
extract_model(P){
01:  for i=1 in (length(P)-1) do
02:     A = elements_in(P);
03:     if effect(A[i]) ∩ precond(A[i+1]) ≠ ∅
04:     then
05:        M := M ∪ pair(A[i],A[i+1]);
06:     else
07:       k = i + 1;
08:       while k <= length(P) do
09:         k = k + 1;
10:         if effect(A[i]) ∩ precond(A[k]) ≠ ∅
11:         then
12:              M := M ∪ pair(A[i],A[k]);
13:              exit_while;
14:         end if
15:       end while
16:
17:       h = i + 1;
18:       while h > 0 do
19:         h = h - 1;
20:         if effect(A[h]) ∩ precond(A[i+1])≠∅
21:         then
22:              M := M ∪ pair(A[h],A[i+1]);
23:              exit_while;
24:         end if
25:       end while
26:
27:     end if
28:  end for
29:  return M
}
```

Fig. 5. Algorithm: Model Generation

After dealing with A_2, the algorithm returns to the main loop to evaluate A_i where $i=3$. In this case, A_i corresponds to C. Next, the pair (A_i , A_{i+1}), which corresponds to (C, D), is checked for ordering. The pair (C, D) is added to M because the C effects are in the D preconditions. The activity D being the last in line cannot establish any new links. Therefore, the main loop is concluded and the algorithm is finished. The result is a model M represented by the set $\{(A, B), (B, D), (A, C), (C, D)\}$.

Conditional Flow

Planning is more difficult in processes that include activities with conditional ramifications. This is so because a classical planner generates just one linear plan that satisfies the goal. It does not take into account conditional ramifications.

The strategy we use to overcome this problem is described as follows. Firstly, the activities without conditional effects are taken to a planner. Secondly, the plan is searched from the beginning to the end in order to find a subactivity. If the algorithm succeeds, the subactivity is removed and the planner is called into action again. In this second execution, it is possible to find or not a plan. A plan is not generated if all

possibilities of execution were already tried. This process of subactivity deletion and plan generation is repeated until no new plan is generated.

The result of the algorithm described above is a set of plans $CP=\{P_1,...,P_n\}$, to be used in the generation of the process model.

4 Results and Discussion

In order to test the efficiency of our planner, we compared its performance with POND, a state of the art planner. The experiments were done in an Athlon XP 2000+ computer equipped with 700MB memory, and 1.67 GHz frequency. We have done two experiments.

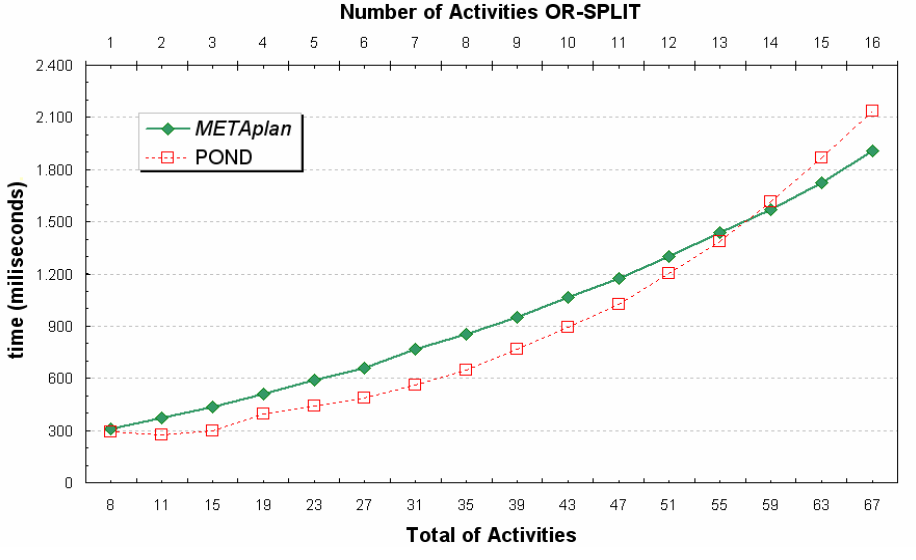


Fig. 6. Result of first experiment: Execution Time by number of activities *OR-SPLIT*

In the first experiment, the goal was to measure the performance of these two planners in problems with few sequential actions and different number of conditional actions. The result of our first experiment is depicted in Figure 6. Notice that after a number of conditional actions (approximately 14 conditional actions) POND takes more time to return a plan. We also noticed that POND ended up in an infinite loop when one of the branches in an OR-Split node did not lead to a goal. This did not happen with METAPlan using FF.

In the second experiment, we worked with the same problems. However, we increased the number of sequential actions. We added up 46 sequential actions. The result is presented in Figure 7. Again, the performance of METAPLAN is higher than POND.

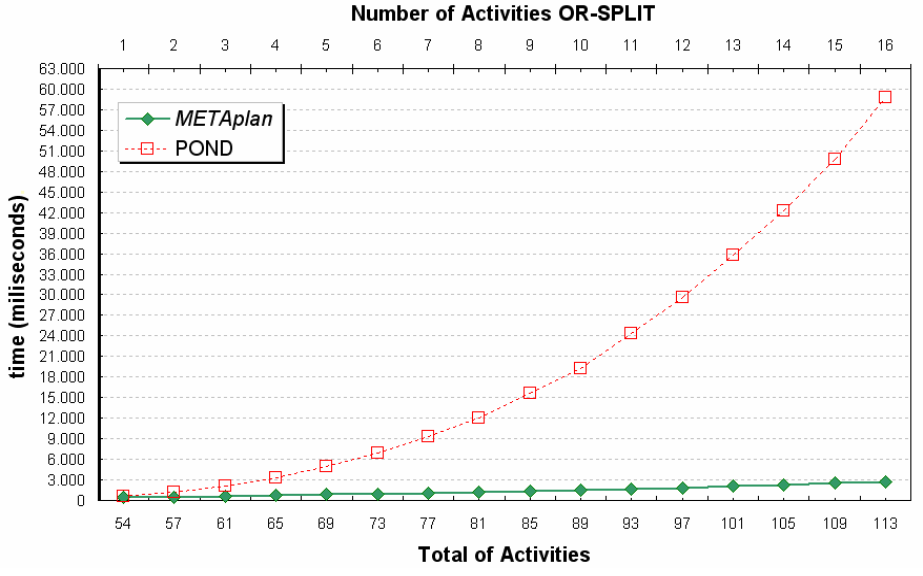


Fig. 7. Result of second experiment: Execution Time by number of operators

5 Related Work

In 1976, Warren introduced Warplan-C[12] the first conditional planner. It was a linear planner using STRIPS rules extended to express uncertain outcomes, designed to support automatic programming. The system was purely forward-planning. Essentially, the idea here is that certain actions may have several alternative outcomes. It could handle actions that had just two possible outcomes, P or ¬P, and did not merge the resulting plan branches. During the first planning pass, the planner assumes that all of the conditional actions are unconditional with a single outcome P. Then, it tries to develop a plan using the other branch of each conditional action.

POND[13] is a state-of-the-art contingent planner that descends of the lineage of Graphplan-based planners. The first member that kind was SGP, which used multiple planning graphs, one for each possible initial world, keeping track of the relationship between these graphs. POND replaces the use of multiple planning graphs with a single planning graph in which propositions/actions are labeled with formulas describing the initial worlds under which they can be true/can be applied.

Contingent-FF[14] is another effort in this area. It is an extension of Conformant-FF to further treat domains with partial observation, finding tree-shaped plans with branches. It also includes a preliminary treatment of a simple form of non-deterministic effects.

The work developed here was built up on Warplan-C. Besides allowing the use of different classical planners, our planner extends Warplan-C by generating plans with points of convergence and partial ordering of actions.

6 Conclusion and Future Work

In this paper we described a simple algorithm for contingent planning. A main goal was to take advantage of recent algorithms for classical planning. Despite its simplicity, the algorithm achieved a good performance. However, a detailed analysis of its complexity and more experiments should be done.

As we stated earlier, we are investigating the use of planning in workflow modeling and enactment. In workflow modeling it is possible to describe loops of activities. It is not common to find loops of actions in the planning literature. In this way, an avenue of research is the development of mechanisms that allow to incorporate loops in our planning algorithm.

References

1. Russell, S. e NORVIG, P. Artificial Intelligence – A Modern Approach, 2. ed. Prentice Hall (2003).
2. Pryor, L. and Collins, G.: Planning for Contingencies: A Decision-based Approach. Journal of Artificial Intelligence Research, Vol. 4 (1996), pp. 287-339
3. Penberthy, J. S.; Weld, D.: UCPOP: A sound, complete, partial order planner for ADL, In: Proceedings Third International Conference on Principles of Knowledge Representation and Reasoning, (1992) p.103-114
4. 4. Weld, D.; Anderson, C. and Smith, D.: Extending Graphplan to Handle Uncertainty and Sensing Actions, AAAI (1998), 897-904
5. Blum, A. and Furst, M. L. Fast planning through planning graph analysis. Proceedings. IJCAI-95, Montreal, Canadá, (1995).
6. Peot, M. and Smith, D.: Conditional nonlinear planning. In Proc. of 1st Int. Conf. on AI Planning Systems (AIPS'92), (1992) pp. 189--197
7. Goldman, R. P., and Boddy, M. S.: Conditional linear planning. In Proc. Second Int'l Conf. on Artificial Intelligence Planning Systems, (1994), 80--85
8. Etzioni, O.; Hanks, S.; Weld, D.; Draper, D.; Lesh, N. and Williamson, M..: An Approach to Planning with Incomplete Information. In Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning, (1992), pp 115-125
9. Fikes, R. E.; Nilsson, N. J.: *STRIPS:* A new approach to the application of theorem proving to problem solving, Artificial Intelligence 2, (1971) p. 189–208
10. McDermott, D., et al. *PDDL* - The Planning Domain Definition Language, Technical Report CVC TR-98-003 / DCS TR -1165, Yale Center for Communicational Vision and Control, 1998
11. WFMC - Workflow Management Coalition, 2004, "The workflow reference model", http://www.wfmc.org, 2004, June.
12. Warren, D. H. D. Generating Conditional Plans and Programs. In Proceedings of the Summer Conference on Artificial Intelligence and Simulation on Behavior, (1976). 344-354
13. Bryce, D.; Kambhampati, S.: Cost Sensitive Reachability Heuristics for Handling State Uncertainty. In: 21st Conference on Uncertainty in Artificial Intelligence. University of Edinburgh, Edinburgh, Scotland, 2005
14. Brafman, R.; Hoffmann, J. Conformant Planning via Heuristic Forward Search: A New Approach. In: KOENIG, S.; Zilberstein, S.; Koehler, J. Proceedings of the 14th International Conference on Automated Planning and Scheduling. ICAPS-04, Whistler, Canada, (2004) p. 355-364

Assessing the Value of Future and Present Options in Real-Time Planning

Emmanuel Martinez, Ramon F. Brena, and Hugo Terashima-Marín

Center for Intelligent Systems, Tecnologico de Monterrey
A00588294@itesm.mx, ramon.brena@itesm.mx, terashima@itesm.mx

Abstract. Highly dynamic environments with uncertainty make inadequate long or rigid plans, because they are frequently dismissed by the arrival or new unexpected situations. In these environments, most approaches eliminate planning altogether, and evaluate just the current situation. We are interested in on-line planning, where execution and planning are interleaved, and short plans are continuously re-evaluated. Now, the plan evaluation itself could be an important issue. We have proposed in our recent work to evaluate plans taking into account the quantity and quality of future options, not just the single best future option. In this paper we present a detailed evaluation of real-time planning performance, changing the importance given to the current situation, to the best future option, and to the set of future options respective evaluations, in the context of the simulated soccer Robocup competition. Our results show that a well-tuned combination of the mentioned factors could outperform any of them alone.

1 Introduction

Real-time soccer is a complex game, involving a great deal of coordination between team members, and the development of strategies leading to marking as many goals as possible and receiving as few goals as possible.

Robocup simulated soccer represents that complexity to a certain degree, meaning that many interesting complexities are present in the simulated soccer as provided by the soccerserver [4], like limited vision and hearing, noise, etc.

Thus, simulated soccer is a challenging problem from the standpoint of coordination, communication, uncertainty, learning, planning, and so on [9].

Many Robocup teams have applied techniques issued from the Artificial Intelligence (AI) field [14] to develop sophisticated skills. Among those AI-related techniques we find Neural Nets, Reinforcement Learning and Probabilistic Reasoning for low-level skills [15], and decision trees [16], Reinforcement Learning [17] and Multiagent Coordination methods like Coordination Graphs [11], for high-level skills.

One area that perhaps has not reached a high degree of development in Robocup is planning [1]. Due to the very dynamic nature of simulated soccer, long-term planning is pointless, as almost any long plan will fail when facing unexpected conditions. Thus, most teams are highly reactive and rely on very

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 522–531, 2006.

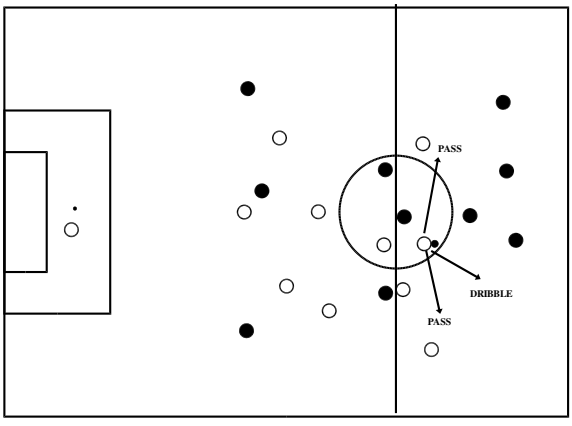


Fig. 1. Example of the problem of decision making in Robocup

polished low-level skills like ball interception and shooting more than on clever playing ideas. Very few teams have applied a planning method to Robocup; in [13], a planning method using opponent modelling is applied for the Coach league team from Carnegie Mellon University. Using bayesian networks, the coach models the opponent's behavior to generate a plan. Finally, the coach communicates the plan to the players. In [6] players have different plans in their memory and they search for one plan to score; their plans consist of dribbles and passes. The cited works use traditional planning methods where the starting point and the ending point of the plan are defined before its execution starts.

In most planning methods [18], when a plan is interrupted at execution time due to unexpected events, players have to start over from scratch a new plan -which will be most probably interrupted as well.

Let us examine an example. Fig.1 shows a game situation where the agent with the ball must take a decision between giving the ball to a well-positioned, but lonely partner, or filtering a long pass to the right, or even to continue dribbling, as indicated by the arrows in the figure. We can see that after passing to the left, the lonely player will have almost no options other than advance and shoot, but could be blocked by many enemies in that part of the field. If instead it dribbles to the right, it still has the other options. The point here is not that dribbling is better than passing, but that under uncertainty it is good to have options for cases where what seemed to be the best option just fails and has to be dropped. In our method we stress the importance of having as many open *options* as possible.

The basic idea in our planning method is to take into consideration, when evaluating a particular possible move, how many and how good *options* the action keeps open after executing it. The advantage of doing this, instead of just selecting the best possible plan or action at a given moment according to some metric, is that in a highly dynamic environment like Robocup, plans often get stuck due to unexpected events, so the agent has to replan from scratch. In our

system, on the contrary, when a plan gets stuck, it is more probable that there is another remaining option -precisely because the planning method has been fostering good options all the time.

Our planning method belongs to the class of on-line planning [8]. On-line planning interleave planning and execution whereas in traditional planning a plan is made and then executed. Agent-centered search [10] is a technique that implements on-line planning, restricting planning to the part of the domain where the current state of the agent is found. Agent-centered search decides on the local search space, searches an action, and executes it. This process is repeated until the goal state is reached.

In the next section we will detail our planning method. In section 3 we will describe the Robocup application of our method in our Robocup team, then we present some experiments supporting our claims, and finally in section 5 we will discuss pros and cons of our planning method, list some results of the team, and draw some future work lines.

2 Method Description

Our planning method considers a collection of current possible actions and the next cycle future actions. [1] Let a specific play (like pass, clearball, etc.) be a triple (p, π, τ), where p is a play identifier, and π and τ are lists of parameters, π for player identifiers and τ for numeric parameters. Each τ_i has either a numerical value or the "undefined" value $\perp$. A parameter with value $\perp$ is said to be "uninstantiated". A similar convention is used for π. For instance, a pass is a triple $(pass, [passer, receiver], [direction, force])$. Partial instantiation is possible, for example, the parameters *passer* and *receiver* could have a value, but not *direction*. A "playbook" [2] P is just a set of all uninstantiated plays.

The collection of current possible actions (CPA) is a subset of the playbook P where the π parameters are instantiated to team members identifiers. This represents the plays that are supposed to be applicable to the current situation. Of course the CPA could have every possible instantiation of every play in P, but we use heuristics to reduce its size. This is equivalent to assign to each play in P a precondition that should be satisfied in order to appear in the CPA. Plays satisfying these heuristics are called "plausible".

Preconditions are of two kinds:

- Static preconditions, which refer to the situation in the playing field, like the proximity of enemies and teammates; for instance, we can avoid passes to teammates not in the neighborhood.
- Sequencing preconditions, which refer to the possibility of executing two given plays in sequence. We say that two plays p_1 and p_2 are *compatible* if p_2 can "reasonably" follow p_1, which is determined by heuristics. For instance, after a "clearball" it is not reasonable to play a dribble, as the ball will be very far from the given player.

[1] We decided not to consider more than one future cycle, so our planning trees are just two levels deep, see section 5.

The playbook have the following offensive plays: [2] Pass, Filtered Pass, Dribble, Outplay (a very long dribble), Clearball (a kick to certain position), and Shoot to goal. [3]

We use a *search tree* for evaluating the playing options, defined as follows:

- The root R represents the current situation.
- For each plausible partially instantiated play p_i from the playbook P, create a branch leading to a son labeled with the action p_i. We denote by $s_i(p)$ the i-th successor of play p in the tree.
- The preceding step is repeated for the sons, giving a two-level tree. [4]

Of course the search tree defined above is classical of AI methods, but we use an original way of evaluating the tree's nodes. Each action p_i of the tree will be associated to a numeric value $E(p_i)$, as described in the following.

Our evaluation of possible plays is based on their *expected utilities*, i.e. the product of their benefit (in case they are successful) by their probability of success; this is why we call it *combined evaluation*:

$$e(p) = bf(p) * pf(p) \tag{1}$$

where bf is the *evaluation of benefit* function (a heuristic taking into account the position of ball and players, see [3]), and pf the *feasibility* function, which returns a number between 0 and 1. The feasibility function is supposed to correspond to the fraction of times a given play could be successful in the given situation.

The original idea we present consists of evaluating actions not only using the best-evaluated son, like in minimax procedures, but a combination taking into consideration the accumulated benefit of all the future actions that will be possible to execute after the current one. The result of this is to give some weight to the quantity and quality of future options generated by a given action.

The formula used to calculate the combined evaluation $E(p)$ of a first-level play p is as follows:

$$E(p) = pf(p)(k_1 bf(p) + k_2 \max\{e(s_i(p))\} + k_3 \frac{\sum\{e(s_i(p))\}}{n_p}) \tag{2}$$

where n_p is the total number of successors of action p, $e(s_i(p))$ are the evaluations with respect to the feasibility and utility of the future actions $s_i(p)$ of the current possible action p, for $i = 1 \ldots n_p$, and finally k_1, k_2 and k_3 are constants for fine-tuning the relative importance of the three terms in the equation's right-hand side.

[2] So far we have limited our attention to the options of the player with the ball. This is not a limitation of our method, but of our current experimentation.

[3] Parameter details for each play are described in [12].

[4] See the discussion section for justification of the two-level tree.

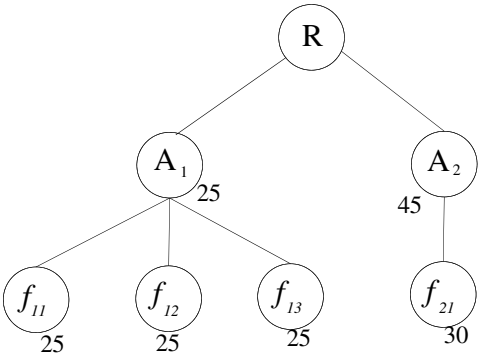


Fig. 2. A simple two-level tree

We illustrate the application of this formula to a simple tree presented in figure 2, where the nodes direct evaluation [5] are written near them. We will assume for this example that $k_1 = k_2 = k_3 = 1$. In conventional maximum-driven tree evaluation, the branch to the right would be selected, either evaluating just the next move, the following one, or both combined. But applying our formula, the evaluations are different. According to our formula the evaluation of node A_1 is:

$$E(A_1) = pf(A_1) * (bf(A_1) + \max\{e(f_{11}), e(f_{12}), e(f_{13})\} + \frac{\sum\{e(f_{11}), e(f_{12}), e(f_{13})\}}{3})$$
$$E(A_1) = 0.9 * (25 + \max\{25, 25, 25\} + \frac{\sum\{25, 25, 25\}}{3})$$
$$E(A_1) = 0.9 * (25 + 25 + 25) = 67.5$$

Whereas the evaluation of the node A_2 is:

$$E(A_2) = pf(A_2) * (bf(A_2) + \max\{e(f_{21})\} + \frac{\sum\{e(f_{21}\}}{1})$$
$$E(A_2) = 0.6 * (45 + \max\{30\} + \sum\{30\})$$
$$E(A_2) = 0.6 * (45 + 30 + 30) = 63$$

So, applying our method, we select the branch to the left, where the action A_1 is chosen to be optimized and then executed.

3 Implementation

Our prototype, the Robocup team described in [12] is based on the UvA Trilearn 2003 team source code [5], on top of which we have implemented some specific skills, like goal-shooting, etc.

In the prototype we also introduced an additional *optimization* step, in which several variations of the action are generated, for instance changing slightly the direction, speed, etc., in order to choose the exact point yielding a maximum utility with respect to its list of future actions. This step is not essential

[5] We are calling "direct evaluation" the assessment of the resulting position itself, without regard to possible future plays.

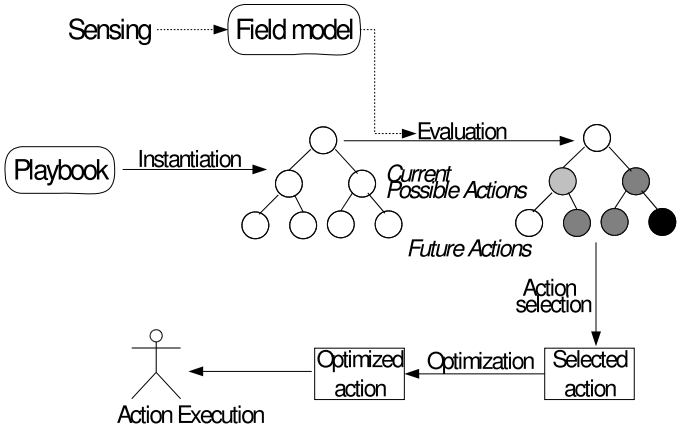


Fig. 3. Robocup prototype architecture

to our planning method we are presenting, and details of it can be consulted in [12].

Figure 3 illustrates the way the planning method is implemented. The basic steps involved in our prototype are:

- First we construct the two-level *play tree* as described above.
- We apply the evaluating procedure described above.
- The best evaluated first-level possible action is chosen from the CPA with its list of future actions.
- The selected first-level action is refined through the optimization process. This is the final decision of the agent.

In the current prototype, our planning method is applied just to the agent with possession of the ball; teammates apply reactive heuristics aiming to help the player with the ball, like to stay far from opponents while attacking, etc. Of course, in principle the planning method could be applied to every single player, and most probably we will do it in future versions (see section 6).

The feasibility function has been adjusted in such a way that it corresponds to the average success rate of plays. Taking a particular play, say "Pass", the prototype calculates the average success of passes throughout an entire game, let it be μ_{real}. Then, we compare it with the average success μ_{pred} predicted by the feasibility evaluation for passes, also during the complete game. They should be equal if predictions are accurate, but of course in practice this is not the case. So, we make an adjustment in the feasibility evaluation for passes, as follows:

$$pf_{new}(pass) = pf_{old}(pass) * \mu_{real}/\mu_{pred}$$

In this way we ensure that at least the average success estimation for passes correspond to the "real" probability, considered as the average success rate. This could be considered as a form of learning.

Another component links our decision mechanisms to a soccerserver command through the use of the UvA Trilearn code [5].

4 Experiments

The most controversial point about our method is that the consideration of future options, as done in 2, indeed improves the performance of a real-time agent or team. We focused the experiments we describe below on proving the relevance of future options consideration.

Taking into account that there are coefficients k_1, k_2 and k_3, we can see that if we make $k_3 = 0$ this accounts as disabling the future options consideration. Thus, we can compare the performance of versions of the team with $k_3 = 0$ and $k_3 \neq 0$, and if the latter has better performance, our point is proven. Methodologically it is important to isolate the variable we want to measure -which is our planning method's performance- from other possible sources of difference.

For the experiments we used the same opponent, which was based on the UvA Trilearn code [5], completed with a decision tree reimplemented from the description in the cited report. With the use of the proxyserver package [7] we generated statistics to compare the performance of the different versions of the team. For these experiments we eliminated one source of uncertainty by giving to players complete information about their current environment; this is achieved by putting "on" the "fullstate" soccerserver parameter (see discussion).

We ran a total of 350 matches, varying the coefficients k_1, k_2 and k_3. The results are shown in table 1 where each row presents the results of 50 matches. The rows are labeled as $k_1k_2k_3$, so for instance 011 means that $k_1 = 0$, $k_2 = 1$, and $k_3 = 1$. As explained above, making a parameter zero is equivalent to dismissing the corresponding component of play evaluation.

The statistical parameters in the table are the following:

- Total goals scored (GS in the table)
- Goals received (GR)
- Earned Points (Points)
- Percentage of possible points (% Points)
- Average Goal Difference (AGD), with its Standard Deviation, Max (Average plus Std. Dev) and Min (Average minus Std. Dev.)

We can see in table 1 that making $k_1 = 0$ produces an extremely poor performance, as consideration of current position evaluation is dismissed. This applies to the first three rows, which we will disregard in the following. The most interesting rows in the table are from forth to seventh. In particular, if we compare row 5th against 4th, we can see a substantial increment in goals scored[6] when $k_3 = 1$, meaning that future options consideration has an important impact on performance. This is reflected also in other statistical parameters, like average goal difference, etc.

[6] Goals received are far less important, because the planning method is executed by players only when attacking.

Table 1. Experimental results

Team	GS	GR	Points	% Points	AGD	Std. Dev	Max	Min
001	1	11	41	27.33	-0.20	0.45	0.25	-0.65
010	0	4	46	30.67	-0.08	0.27	0.19	-0.35
011	0	6	44	29.33	-0.12	0.33	0.21	-0.45
100	60	8	109	72.67	1.04	1.21	2.25	-0.17
101	84	9	119	79.33	1.50	1.50	3.00	-0.00
110	82	4	126	84.00	1.56	1.36	2.92	0.20
111	97	5	140	93.33	1.84	1.17	3.01	0.67

A similar comparison can be made between rows 7th and 6th, where a consistent performance improvement is achieved by the team 111 when compared with team 110. As the only difference between those teams is the value of the independent value in the experiments, which is in this case k_3, it is valid to attribute to it the performance difference.

5 Discussion

The main purpose of this paper is to clearly demonstrate, from rigorous experiments, that in real-time planning the consideration of future options can have a great impact on performance. We do not claim that our team is any better than other teams which do not consider future options, because there are many issues that could account for a better performance, particularly low-level skills, which we do not think are very well developed in our Robocup teams. Further, there are many issues that could have impact in our planning method performance, like the accuracy of evaluation functions. Indeed, bad estimations of either utility or probability would lead to completely erroneous play evaluations. Actually we had to work a lot on these functions before the planning method delivered acceptable results, and yet we have much work to do refining those functions But despite the mentioned limitations, we are confident about the soundness of our experiments to prove the point we wanted to prove.

Note that in our experiments we left the soccerserver parameter "fullstate" on, which means that complete information about the current situation is given to players by the soccerserver. We did so, at least for basic experimentation purposes, because inaccuracy in modeling the current situation would affect as "noise" the plays' evaluation by formula 2. We plan to investigate the application of the planning method with more uncertainty involved (see section 6).

Formula 2 itself could need some adjustments for a specific application; for instance we started with a version where the third term in the right-hand side was $k_3 \sum\{e(s_i(p))\}$, that is, taking the sum instead of the average, but experimental results were much weaker. The explanation could be that in Robocup to have too many weak options is not convenient, so dividing by the number of options made the correction. We think similar adjustments could be necessary for other applications.

One potential disadvantage or our method is its high computational cost, because we need to perform many evaluations and then one optimization –involving even more evaluations. We relied on heuristics to reduce the search space for making the complexity manageable.

6 Conclusions and Future Work

We have presented and evaluated a new real-time planning method for highly dynamic domains. In our method, possible actions evaluation is based not only on the best evaluated move in the search tree, but on a metric considering the quantity and quality of the future options left available by the actions being considered.

In this paper we report an implementation of our method in the domain of the Robocup competition basically for the sake of performing rigurous experiments validating our claim that a performance improvement can be achieved by taking into account future options, as they appear in formula 2.

We present experiments that clearly show that our planning method outperforms traditional (maximum immediate evaluation) planning.

Current research is focused on carefully refining utility and feasibility functions. We plan also to provide better field modeling for keeping track of the current situation without needing the "fullstate" parameter on. Future research will include adapting our planning method to applications outside the Robocup competition, like electronic auctions, client-driven dynamic manufacturing, and aerial combat real-time decisions.

Acknowledgements

This work was supported by the Monterrey Tech's Research Grant CAT011. We would like to thank the anonymous reviewers for their insightful comments.

References

1. James Allen, James Hendler, and Austin Tate. *Readings in Planning*. Representation and Reasoning Series. Morgan Kaufmann, San Mateo, California, 1990.
2. Michael Bowling, Brett Browning, and Manuela Veloso. Plays as effective multiagent plans enabling opponent-adaptive play selection. In *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04)*, 2004.
3. R. Brena and E. Martinez. Maximizing future options: An on-line real-time planning method. *Lecture notes in computer science*, 3789:970–979, 2005.
4. Mao Chen, Klaus Dorer, Ehsan Foroughi, Fredrik Heintz, ZhanXiang Huang, Spiros Kapetanakis, Kostas Kostiadis, Johan Kummeneje, Jan Murray, Itsuki Noda, Oliver Obst, Patrick Riley, Timo Steffens, Yi Wang, and Xiang Yin. Users manual: Robocup soccer server (for soccerserver version 7.07 and later).
5. R. de Boer and J. Kok. The incremental development of a synthetic multi-agent system: The uva trilearn, 2001.

6. Ahmad Farahany, Mostafa Rokooey, Mohammad Salehe, and Meisam Vosoughpour. Mersad 2004 team description, 2004.
7. Ian Frank, Kumiko Tanaka-Ishii, Katsuto Arai, , and Hitoshi Matsubara. The statistics proxy server. In Peter Stone, Tucker Balch, and Gerhard Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 303–308. Springer Verlag, Berlin, 2001.
8. Lise Getoor, Greger Ottosson, Markus Fromherz, and Bjoern Carlson. Effective redundant constraints for online scheduling. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)*, pages 302–307, Providence, Rhode Island, July 1997. AAAI Press / MIT Press.
9. Frans Groen, Matthijs Spaan, and Nikos Vlassis. Robot soccer: Game or science. In *Proceedings CNR-2002. Editura Universitaria Craiova*, October 2002.
10. Sven Koenig. Agent-centered search. *AI Magazine*, 22(4):109–131, 2001.
11. J. Kok, M. Spaan, and N. Vlassis. Non-communicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems*, 50(2-3):99–114, 2005.
12. Emmanuel Martinez. A real-time planning method for Robocup (in spanish). Master's thesis, Tecnologico de Monterrey, Mexico, 2005.
13. Patrick Riley and Manuela Veloso. Planning for distributed execution through use of probabilistic opponent models. In *IJCAI-2001 Workshop PRO-2: Planning under Uncertainty and Incomplete Information*, 2001.
14. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.
15. Peter Stone. Layered learning in multiagent systems. In *AAAI/IAAI*, page 819, 1997.
16. Peter Stone and Manuela Veloso. Using decision tree confidence factors for multiagent control. In Katia P. Sycara and Michael Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 86–91, New York, 9–13, 1998. ACM Press.
17. Peter Stone and Manuela M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.
18. M. Zweben and M. S. Fox. *Intelligent Scheduling*. Morgan Kaufmann, 1994.

Reading PDDL, Writing an Object-Oriented Model

Flavio Tonidandel[1], Tiago Stegun Vaquero[2], and José Reinaldo Silva[2]

[1] Centro Universitário da FEI
IAAA Lab – EE - São Bernardo do Campo, Brazil
[2] Escola Politécnica – Universidade de São Paulo
Design Lab. – PMR – São Paulo, Brazil
flaviot@fei.edu.br, tiago.vaquero@poli.usp.br,
reinaldo@usp.br

Abstract. There are many efforts towards a combination of planning systems and real world applications. Although the PDDL is in constant evolution, which improves its capability to describe real domains, it is still a declarative language that is not so simple to be used by the non-planning community. This paper describes a translation process that reads a domain specification in PDDL and transforms it into an object-oriented model, more specifically into a version of UML for planning approaches. This translation process can let a designer read PDDL domains and verify it with some powerful tool like itSIMPLE or GIPO, or it can allow a planning system that only reads object-oriented models to run in domains described in PDDL originally.

1 Introduction

The PDDL (Planning Domain Description Language) [3] is a standard language to describe planning domains and problems since 1998 and it becomes a reference language for planning and non-planning community committed to present new models for real planning domains. However, as highlighted in some works [1], the PDDL is not intuitive enough to be an easy-to-use language. On the contrary, it has become more and more complex whereas new versions are released.

Concerning on the complexity and the lack of a tool that can support knowledge engineering processes, the first international event that focus in this subject was the ICKEPS 2005 (International Competition on Knowledge Engineering for Planning and Scheduling) where new proposals for tools and modeling languages emerged as well as the discussion about solving real planning problems.

On ICKEPS 2005, two works on object-oriented languages appeared: itSIMPLE [7] and GIPO [5]. They showed that they are promising and valuable tools for modeling, verification, validation and analysis of planning domains.

In fact, the object-oriented model is more intuitive for planning designers than PDDL. Since the planning community intends to build a bridge over the gap between real applications and planning simulation domains, it is suitable to have some flexibility in the modeling language as well as an easy-to-use processing model.

Many domains have been described in PDDL and many planners can read only domain and problem specifications in PDDL. The GIPO tool, as well as itSIMPLE,

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 532 – 541, 2006.

can export their models to PDDL. However, there is only one work that translates PDDL into an object-oriented model [6] that is very specific for OCLh language.

This paper aims to define a translation process that can read a PDDL model and write an object-oriented model in a general way, more specifically into UML.P (UML in a Planning Approach). This paper is the first attempt to perform this translation and, therefore, it just considers the classical domain described in STRIPS-like PDDL.

The translation process was implemented and tested in the itSIMPLE tool, and this process can be adapted in the future to generate object-oriented models for any tool.

2 Overview of UML for Planning

The UML (Unified Modeling Language) [2] is one of the most used languages to model a great variety of applications and we believe that most engineers in several application areas are somehow familiar with the UML language [4]. The purpose of UML is to be a general modeling language, thereby we will consider the UML in the Planning Approach, which will conveniently be called UML.P throughout this paper.

2.1 Class Diagram

The class diagram is a representation of the planning domain structure showing the existing entities, their relationships, their features, methods (actions) and constraints.

In order to simplify the planning modeling process, *UML.P* proposes a General Class Structure composed by *Agent* class and a domain *Environment* class. The *Agent* changes the arrangement of objects, which compose the environment.

In the class diagram, there are relationships between classes where any relationship or dependency between two or more classes is an association. There are three kinds of relations: Association, Aggregation and Composition.

Associations just connect classes identifying some appropriate meaning, for instance, association “at” between the class *Vehicle* and *Location* in the Logistic domain. Each association between entities has its respective multiplicities. For example, one or many trucks (1…*) can be at one and only one place at a time.

Aggregation and Composition are a special kind of association which can be read as a “have”, “belong” or “made of”. For this paper, we will consider only Aggregations and Associations.

In *UML.P*, every class that can perform actions in the domain will be a specialization (subclass) of the class Agent, for convenience. These classes with dynamic behavior can have methods (actions) defined in class specifications. All the others classes will be associated to the Environment with an aggregation relation. The StateChart Diagram models the dynamic behavior of action.

2.2 StateChart Diagram

The StateChart Diagrams are very useful to represent entities that perform dynamic behavior. All methods (actions) in the class diagram are specified in these diagrams.

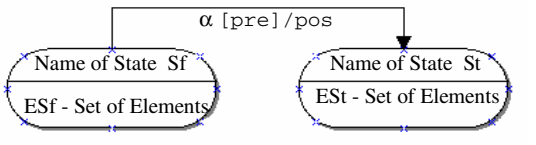


Fig. 1. The General Structure of the StateChart Diagram. The `pre` and `pos` are sets of elements characterizing preconditions and post-conditions of the action α. On the right there is an example of a Package StateChart Diagram of the Logistic Domain.

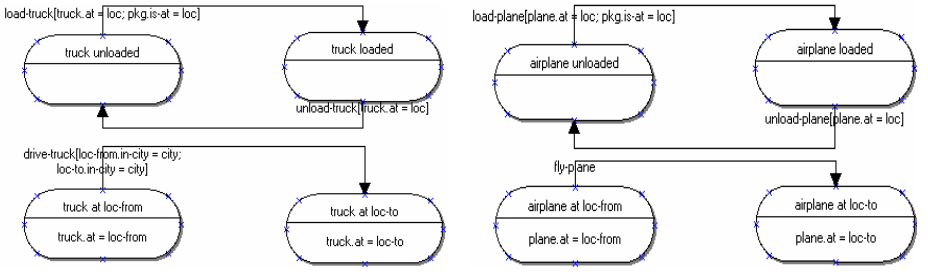


Fig. 2. Truck and Airplane StateChart Diagrams modeled in *UML.P*

Any class in Class Diagram can have its own State Diagram. A StateChart Diagram is unique for a class and it does not intend to specify all changes caused by an action, but only specify dynamic behaviors for an object of the class caused by one or more actions. This dynamic behavior is expressed by actions that affect the objects of the class. An action (method), therefore, can appear in many StateChart Diagrams.

For the planning context, the *UML.P* has special remarks during StateChart diagram construction:

- There is one diagram for each class capable of dynamic behavior. For instance, the truck class in the Logistic domain is a class with dynamic behavior;
- The state elements in the diagram represent the possible states of an object. The state is defined by the main values of its attributes.
- The state elements are not necessary mutually exclusive;

Summarizing, the states in StateChart Diagram describe the possible states of the class. The actions are transitions among these states. The general structure of a StateChart Diagram is shown by figure 1.

In a StateChart Diagram of a specific class, we have some considerations. First, the states lie about an object of the class that directs performs the action. The action has its own pre and post definition (as in figure 1) described explicitly in the diagram. These pre and post definitions can lie about objects of the StateChart Diagram´s class or another class in Class Diagram that can receive or suffer an indirect effect of this specific action (transition) but does not have a StateChart Diagram.

The structure in figure 1 has two states (named Sf and St) that are considered the State-form and State-to of the action α. Esf are Elements of the state Sf , similar for ESt. It is important to realize that when the action α is performed, the ESf elements

become false and the elements of ESt become true. For convenience, the elements of ESf and Est lie about the objects of the StateChart Diagram´s class.

```
Function translation_process(PDDL: PDDL file)
  Extract_types(PDDL);
  Create_specializations from types ;
  Create_associations(PDDL);
  For each action α in F:
       ABS_precond(α) and ABS_effect(α);
  Endfor
  Gather_States ;
  For each class in Class Diagram   Create_StateChartDiagram;
  Endfor.
```

Fig. 3. General Algorithm for the translation process

Considering and example, since the classical Logistic Domain has only two entities with dynamic behavior (*Truck* and *Airplane*) and one entity that can receive effects of actions (*Package*), it is necessary three StateChart Diagram. Figure 1 focused on the Package, Figure 2 focused on the class Truck and the class Airplane.

Observe that in the Plane StateChart Diagram, for instance, the States lie about the plane class and the pre and post definitions lie about an indirect effect for the plane (`plane.at=loc`) and another indirect effect of the transition between loaded and unloaded that is `pkg.at=loc`.

3 PDDL Translation into Object-Oriented Model

The translation of PDDL into an object-oriented model is a process that must make some semantic considerations in PDDL description and that will provide an elegant way to describe PDDL domains as object classes and transition of states.

The following section will describe the translation process used by the itSIMPLE tool. It translates each peace of a PDDL domain into each Diagram of the *UML.P*.

The figure 3 shows the entire process of the translation to *UML.P* from PDDL in a higher level algorithm language. Definitions for each function described in the figure 3 will be in the following sections.

3.1 Reading PDDL and Creating Class Diagram

The first step is to create a Class Diagram in *UML.P* by identifying classes in the PDDL specification. The PDDL has a structure of types that appears in the domains with a tag `:typing` in the requirements. In fact, there is a semantic correspondence between types in PDDL and classes in *UML.P*.

Therefore, the translation process, through the function **Extract_types(PDDL)** just reads the types in a PDDL file (F) and creates the classes in *UML.P*.

However, in order to define which class (environment or agent) will be a super-class of a given one, some analyses are necessary under the following definition:

Definition 1 (action structure). In PDDL, an action α has the following structure:

```
(:action α
  :parameters (?v1 - tp1 ... ?vn - tpn)
  :precondition ρ
  :effect (and  δ τ ))
```

Where p_i = $?v_i$ - tp_i is the i^{th} parameter of the action; δ is the delete list (negative effects) and τ is the add list (positive effects) of the action.

Aiming the translation to object-oriented model, some consideration must be made because PDDL action definition has no information about the most important object of the action. In fact, there is no imposed rule for definition of action parameter order in PDDL. We will call the order of the parameters in the :parameter tag in PDDL as a parameter list (PL). Formally:

Definition 2 (Parameters List). The Parameters List, denominated PL, of an action α is a list that takes into account the given order (from left to right) of the parameters in the :parameter tag in PDDL.

We will consider, therefore, the order of the PL as an indicative of the importance of each type (class) in the action definition. For instance the PL of following parameters of action Load-Truck (:parameters (?truck – truck ?pkg - package ?loc - place)) would be PL = {truck: truck, pkg: package, loc: place}.

The most important class of the action Load-Truck above is the truck, that is the first class of PL. With the consideration above, we can define the class where each action will be specified in the Class Diagram as methods.

The Class Diagram has a main structure that incorporates two default classes: *Agent* and *Environment*. Any action defined in PDDL will be allocated as a method of the class of its first parameter in PL and this class will be a subclass of *Agent*.

Besides Classes we have associations and aggregations in the Class Diagram. The aggregations are extracted directly from type definitions in PDDL by the function by also creating specializations through the function called **Create_specializations.** For example, truck is a vehicle in PDDL (described as truck – vehicle) and in the Class Diagram there will be a specialization relation between class Truck and class Vehicle.

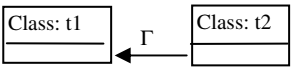
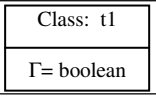
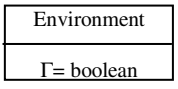
The associations are relations between classes with a name and multiplicity. These associations correspond semantically to a predicate of two arguments in the PDDL description. For example, *in(package, airplane)* means an association, called *in*, from package class to airplane class. The function **Create_associations(PDDL**) creates the respective associations under the following definitions:

Definition 3 (predicate structure). A predicate Γ in PDDL has the following structure: (Γ [?v1 – tp1] [?v2 – tp1]) where [ ] indicates optional argument, vi indicates variable i and tpi indicates the type of the variable i.

The predicates describe relations between classes. The main class of the predicate is the first argument of the predicate. For example, in the predicate (at ?truck1-truck ?city1-city) means that the predicate main argument is the truck1.

Concerning the number of arguments (arity) of the predicates, we can consider P_0 as a Predicate without argument (arity = 0), P_1 as a predicate with one argument (arity = 1) and P_2 as a predicate with two arguments (arity = 2).

Table 1. The correspondence between PDDL and UML for predicates

Predicate in PDDL	UML description	Graphical UML
(Γ ?v1 – tp1 ?v2 –tp2) Arity/2	Γ.v1 = v2	
(Γ ?v1 – tp1) Arity/1	Γ.v1 = true or Γ.v1 = false	
Γ Arity/0	Γ = true or Γ = false	

We do not consider predicates with more than 2 arguments although it can be expressed in UML. Table 1 defines the translation function between predicates and associations. Observe that predicates with less than two arguments are described as parameters of a class or the default class environment.

In the translation process, predicates P_1 and P_0 in PDDL are expressed in boolean type in UML.P. Therefore, we can define the negation of a predicate of P_1 or P_0 as:

Definition 4 (Negation of P). For a predicate $P \in P_0 \cup P_1$, $\overline{P}$ is the negation of P by negating the Boolean value of the predicate description. For a predicate $P \in P_2$, in the form p1.v1=v2, $\overline{P}$ will be defined as p1.v1=NULL.

Using the definition above, we can show the following example: if P is "a.p=true" then $\overline{P}$ would be "a.p=false".

In UML.P, there are multiplicity definitions on associations. However, since the PDDL description has no information of quantities between arguments of a predicate, the translation process just considers the multiplicity 0…* for all associations.

3.2 Reading PDDL and Creating StateChart Diagram

In order to understand the semantic relation between actions definitions in PDDL and StateChart Diagrams in *UML.P*, we must realize that the action definition in PDDL is a composition of States (Sf and St) and `pre` and `post` elements for such action in all StateChart Diagrams. In one hand, suppose an action α that goes from Sf to a St in one StateChart Diagram. Predicates that describe Sf are related to precondition and to a delete list of α in PDDL. All predicates of the St will compose the Add list (positive effects) in the effects list in PDDL.

On the other hand, the information in `pre` and `post` that comes together with actions name in StateChart Diagram are posted in the precondition and effects of the action definition in PDDL.

In order to perform the translation of the dynamic behavior of actions, we consider the delete list as a subset of the precondition list. The key to translate PDDL actions into transitions of states is the parameter list (PL) of each action.

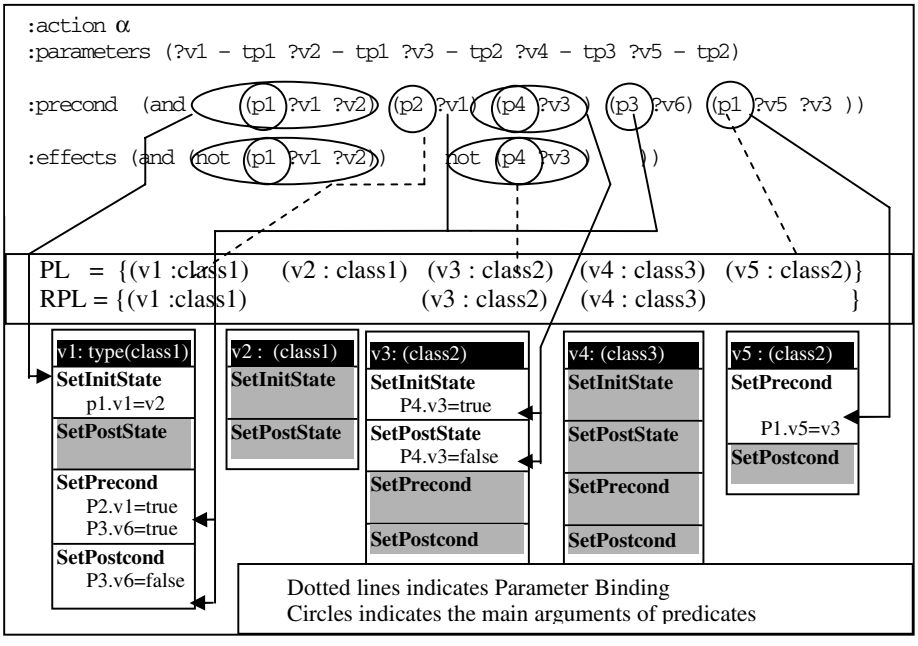


Fig. 4. The general schema of the action analysis and the split of predicates in their correspondent class and set

Considering the predicates in an action, each of them lies to one class or an association. In fact, they are grouped by their correspondent classes.

We must consider that only the first occurrence of each type in the action's parameter list will be considered relevant for action purposes. In order to define relevant classes (types in PDDL) of an action, we will consider the types of the action parameters in PDDL. Since PDDL does not express the relevant types explicitly, we will consider the order of the parameters in an action extracted by the PL list (def. 2).

Then, we will consider the first occurrence of each type in a list of parameters of the action as the relevant reference of that type in such action. In fact, we will create a new list called Relevant Parameters List (RPL). By definition, we have:

Definition 5 (Relevant Parameters List). Let be an ordering of parameters in an action α called PL. The Relevant Parameters List, denominated RPL, is a subset of PL, RPL $\subseteq$ PL, and their elements are the first occurrence of any type in the PL ordered list.

The analysis of actions intends to classify each action in accordance with the parameters in the parameter list (PL). In order to precisely define which class can have a StateChart Diagram, we will define the relevant classes of the parameter list of the action as the classes that can have a StateChart Diagram.

Performing the action analysis, we will consider that each parameter in PL will have two sets of predicates: SetPrecond and SetPostcond. The parameters in the RPL will have two more sets: SetInitState and SetPostState.

```
Function ABS_precond (a: action)
For each P in :precond of a
   i ← Parameter Binding of P;
   if P is a Relevant Predicate and P ∈ δ
        insert P in SetInitState(i)
        If (P ∈ P0∪P1 ) insert  P̄  in SetPostState(i)
   endif;
   if P is a Relevant Predicate and P ∉ δ
        insert P in SetPrecond(i)
   endif;
   if P is not a Relevant Predicate
        insert P in SetPrecond(i)
        insert  P̄  in SetPostcond(i)
   endif;
```

Fig. 5. Algorithm for the ABS structure

The SetInitState is related to Sf in figure 1. In fact, SetInitState of an action is a subset of the Sf elements. Similarly, SetPostState is related to St and SetPrecond and SetPostcond are related to pre and post definitions. This structure of Sets for each parameter in PL is called ABS(Action Behavior Structure).

Therefore, the process of action analysis extracts the predicates that can fit into each set of each parameter. Each predicate will be inserted in a specific set as shown by an example in Figure 4. In order to explain the process in Figure 4, we must define Relevant Predicate and Parameter Binding of a predicate in an action:

Definition 6 (Parameter Binding for predicates with arguments). Consider a predicate $\Gamma \subseteq P_1 \cup P_2$ and its first argument v_1 into the action α definition. The Parameter Binding will be a parameter that is the same variable or value of the first argument v_1.

Definition 7 (Parameter Binding for Predicates with no arguments). Consider a predicate $\Gamma \subseteq P_0$ into the action α definition. The Parameter Binding will be the first parameter of the PL list of the action α.

Definition 8 (Relevant Predicate). A predicate Γ is considered relevant iff its first argument v_1 is in the RPL (Relevant Parameter List).

The most important feature is to identify the parameter in the PL list of the action that each predicate is related to. It is done by considering the first argument of the predicate. This analysis that creates a relation between predicates and parameters of the action is necessary to identify the correct states in the StateChart Diagram.

From now on, we will consider the following notation to denote sets of each parameter of the action:

- SetPrecond(i) and SetPostCond (i) – The Precondition and Postcondition Sets of the parameter i of the PL list.
- SetInitState(i) and SetPostState(i) – The Initial State and Post-State Sets of the parameter i of the PL list. This parameter is a relevant parameter.

```
Function ABS_effect (a: action)
For each P as positive effect in :effects
   i ← Parameter Binding of P;
   if P is a Relevant Predicate
        insert P in SetPostState(i)
        If (P ∈ P0∪P1) insert P̄ in SetInitState(i)
   endif;
   if P is not a Relevant Predicate
      insert P in SetPostcond(i)
      If (P ∈ P0∪P1) insert P̄ in SetPrecond(i)
   endif;
endfor;
```

Fig. 6. Algorithm for the ABS structure

The Figure 4 shows a general schema for the application of the algorithm in figure 5. The algorithm in Figure 5 and 6 shows the process to create the ABS structure.

ABS_precond(α): The algorithm considers the predicates in the precondition list of an action α in PDDL. Each predicate P and its negation $\overline{P}$ can be inserted into the SetInitState or in SetPrecond of the structure of a parameter in PL.

The first operation is to find the parameter 'i' in PL that correspond to each predicate P in precondition. In one hand, if P is relevant and it is in the delete list, it will be added in the SetInitState of the parameter i and $\overline{P}$ will be added to SetPostState if is not P_2.

On the other hand, if P is relevant but it is not in the delete list, P can not be in a State definition, so it will be added in the Setprecond list of parameter i. Since only relevant predicates can be in a state of a class, if P is not relevant, it will only be added to Setprecond of parameter i and its negation will be added to SetPostcond if P is not P_2.

ABS_effects(α): The algorithm is similar to the ABS_precond(α). However, it considers the predicates in the add list and do not compare with the delete list.

After the process of analysis and the extraction of SetInitState and SetPostState of each action, we must gather the states for each class.

Gather_States: The translation process groups by class all SetInitStates and all SetPostStates in a list of states. Then, the process gathers all states which are the same, i.e., that have the same set of predicates.

In order to discover that one state is the same of another state, they must have predicates and variables with the same name and order. Ex: the argument of both on(x,y) in one state and on(x,y) in another state must be equals. If they are not the same, the states will not be comparable and they will be considered different states.

Create_StateChartDiagram: The translation process examines all actions and groups of each class presented in the PL list in order to define the StateChart Diagrams.

For one class, we will have a set of SetIntiStates and SetPostState and the actions related to them. Each state description in SetInitState and SetPostState will become a state in the StateChart Diagram, Sf and St respectively in accordance to figure 1. Then, each action related to these states will be described as a transition between

these states. This action will have incorporated in its `pre` and `post` definition all Setprecond and SetPostcond list of predicates for that class.

This process occurs to all classes and all actions. However, some Setprecond and Setpostcond of some action lie in a parameter that does not have a StateChart Diagram. In these cases, the SetPrecond and SetPostcond will be jointed to the SetPrecond and Sertpostcond of the class of the first parameter of the action.

4 Evaluation

The translation process described in this paper was implemented in itSIMPLE tool. Many STRIPS-like domains like Block World, Logistic, Free Cell, and others were used to evaluate the translation process. In all cases the translation was successfully done and the translation back to PDDL was correctly performed, which certify that the translation process keeps the same semantic in both models.

5 Conclusion

This paper showed a translation process that can read a STRIPS-like domain model described in PDDL and write it in UML.P. The translation process made some considerations in PDDL description because the *UML.P* approach model, which relies on object-oriented specifications, needs more information than the declarative description of the PDDL can provide. However, with these considerations, a model described in PDDL can be easily extracted as object-oriented model and can be read by tools like itSIMPLE to visualize, analyze, and verify the domain description.

References

1. Boddy, M. Imperfect Match: PDDL2.1 and Real Applications. *Journal of Artificial Intelligence Research* 20 (2003) 133 – 137.
2. D'Souza, F. D. and Wills, A. C. *Object, Components, and Frameworks with UML – The Catalysis Approach*. Addison-Wesley. USA and Canada (1999).
3. McDermott, D *et al*. *The PDDL Planning Domain Definition Language*. The AIPS-98 Planning Competition Committee (1998).
4. OMG - Object Management Group. *Unified modeling language specification: version 1.4.* URL http://www.omg.org/uml (2001).
5. Simpson, R.M, McCluskey, T. L., Zhao, W., Aylett, R. S. and Doniat, C. *An Integrated Graphical Tool to support Knowledge Engineering in AI Planning*. Proceedings, 2001 European Conference on Planning, Toledo, Spain (2001).
6. Simpson, R.M, McCluskey, T.L, Liu and Kitchin. "Knowledge Representation in Planning: A PDDL to OCLh Translation", LNCS 1932 (2000).
7. Vaquero, T. S., Tonidandel, F., Silva, J. R. *The itSIMPLE tool for Modeling Planning Domains*. ICAPS 2005 Competition on Knowledge Engineering for Planning and Scheduling, Monterey, California, USA (2005).

A Reactive Lazy PRM Approach for Nonholonomic Motion Planning

Abraham Sánchez[1], Rodrigo Cuautle[1], René Zapata[2], and Maria Osorio[1]

[1] Facultad de Ciencias de la Computación, BUAP
14 Sur esq. San Claudio, CP 72570
Puebla, Pue., México
{asanchez, aosorio}@cs.buap.mx
[2] LIRMM, UMR5506 CNRS, 161 rue Ada 34392,
Montpellier Cedex 5, France
zapata@lirmm.fr

Abstract. This work describes a reactive lazy PRM planner, that integrates the lazy PRM planning approach and the reactive control, using a DVZ (Deformable Virtual Zone). The lazy PRM approach calculates a collision-free and feasible path for the mobile robot before it starts moving under the permanent protection of its DVZ. In the absence of dynamic obstacles, the control is performed by the lazy PRM approach and does not require reflex commands. In the presence of dynamic obstacles in its path, the reactive approach takes the control and generates commands to move the robot away from the intruder obstacles before forcing its DVZ to go back to the original state. Experimental results show the effectiveness of the planner proposed here.

Keywords: Nonholonomic motion planning, deformable virtual zone, lazy PRM.

1 Introduction

The research in robot motion planning can be traced back to the late 60's, during the early stages of the development of computer-controlled robots. Nevertheless, most of the effort has been done recently and has been conducted during the 80's. In the 80's, roboticians addressed the problem by devising a variety of heuristics and approximate methods. Motion planning can be split into two classes: holonomic and non-holonomic motion planning. In non-holonomic motion planning, any path in the free configuration space does not necessarily correspond to a feasible one. Non-holonomic motion planning turns out to be much more difficult than holonomic motion planning. This is a fundamental issue for most types of mobile robots.

From path planning to trajectory control, the motion planning problem for mobile robots has been thoroughly investigated in the case of structured environments. Moving among unknown or badly modeled environments, practically induces the necessity of taking unscheduled and dynamic events into account

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 542–551, 2006.

and reacting as the living beings would do. Therefore, reactive behaviors play a fundamental role when the robot has to move through unstructured and dynamic environments.

Artificial reflex actions for mobile robots can be defined as the ability to react when unscheduled events occur, for instance when they move in unknown and dynamic environments. The problem of designing reflex-oriented artificial systems has been partially solved by using a behavioral approach, which consists in directly relating inputs (stimuli) to outputs (actions) through state machines and to make these elementary machines communicate [1], [2]. These machines are programmed with fuzzy logic, neural networks, deterministic state machines, etc. Another design considers a sensor-based approach that sends feedback sensory information to the robot control loop. The most famous method of this family is the potential method developed by O. Khatib, many years ago [3]. In the first approach, the stimuli are translated into virtual external forces that are simply added to the goal force, when there is a goal; all these forces produce an action, moving the system and locally modifying the world. In the second approach, information provided by exteroceptive sensors, generates the same virtual external forces that contribute to the system evolution, after a comparison with the programmed goal forces.

For the last fifteen years, the scientific community has been interested in the problem of reactive behaviors for collision avoidance in the domain of mobile robots [4], [5]. In this trend, this work aims at providing a practical planner that considers reflex actions and lazy techniques to account for planning with changing obstacles. The paper is organized as follows. Section II presents briefly the Lazy PRM approach. Section III gives an overview of the DVZ principle. Section IV explains the details of the proposed planner. The performance of the planner is experimentally evaluated in Section V. Finally, the conclusions and future work are presented in Section VI.

2 Lazy PRM

In the last decade, much research effort was done on the application of probabilistic roadmap methods (PRM) for different types of motion planning problems [6], [7], [8]. The goal of PRM planners is to capture the connectivity of robot's free configuration space in a finite graph structure (i.e., a roadmap), where each node represents a free robot configuration and an edge between two nodes represents the fact that a simple local planner can find a collision-free path between the corresponding robot configurations. There are two main classes of PRM planners: multiple-query and single-query planners. A multiple-query planner pre-computes a roadmap and then uses it to process many queries [6], [7]. On the other hand, a single-query planner computes a new roadmap for each query [9], [10].

A Lazy PRM approach for nonholonomic motion planning was presented in [10]. The algorithm is similar to the work presented by Bohlin and Kavraki [9], in the sense that the aim is to find the shortest path in a roadmap generated by randomly distributed configurations (in a later work the authors showed that the

use of deterministic sampling improved remarkably past results obtained with random sampling [11]). Once a start-goal query is given, the planner performs A^* search on the roadmap to find a solution. If any of the solution edges are in collision, they are removed from the roadmap and then A^* search is repeated. Eventually, all edges may have to be checked for collisions, but often the solution is found before this happens. If no solution is found, more nodes may need to be added to the roadmap. The most important advantage of this approach, is that the collision checking is only performed when needed. In this case, all edges don't have to be collision checked as in the original PRM case. Experiments show that only a very small fraction of the graph must be explored to find a feasible path in many cases [10]. The Lazy PRM approach provides a kind of missing link between classical grid search and the basic PRM [11]. The spirit is much like classical search (assuming only one execution of the A^* algorithm), but occurs over a randomly or deterministically generated graph, instead of an implicitly defined lattice.

3 The DVZ Principle

The Deformable Virtual Zone (DVZ) approach supposes that the robot is surrounded by a DVZ whose geometry depends on the robot kinematics, and deformations are due to the intrusion of proximity information in the robot's space. The DVZ method consists in minimizing these deformations by locally modifying the control vector(acceleration, orientation, etc) [5].

Definition 1. *For a convex rigid body $R \in \mathbb{R}^3$, we define the* **undeformed DVZ-set** *as any convex surface surrounding R that verifies a one-one correspondence with the boundary ∂R of R.*

Definition 2. *The* **undeformed DVZ** *Ξ_h of a convex rigid body $R \in \mathbb{R}^3$, is the one-one map in the set of convex surfaces, relating ∂R to Ξ_h: It is formally defined by: $M \xrightarrow{\bar{\Xi}_h} P_h$.*

Following the definition, $\bar{\bar{\Xi}}_h = \Xi_h(\partial R)$.

Definition 3. *For a convex rigid body moving among obstacles, with a specific DVZ-set Ξ_h, we define the* **deformed DVZ-set** *$\bar{\Xi}$ as any convex surface corresponding R and included in $\bar{\bar{\Xi}}_h$ that verifies a one-one correspondence with the boundary ∂R of R.*

Definition 4. *The* **deformed DVZ** *Ξ of a convex rigid body $R \in \mathbb{R}^3$, is the one-one map in the set of convex surfaces, relating ∂R to $\bar{\bar{\Xi}}_h$ and formally defined by: $M \xrightarrow{\Xi} P$.*

Definition 5. *The* **deformation** *$\triangle$ is defined as the functional difference of Ξ and Ξ_h:*

$$\triangle = \Xi - \Xi_h \tag{1}$$

The deformation $\triangle$ is a one-one map that associates the vector $P - P_h$ to the point $M \in \partial R$. It can therefore be considered as a vector field defined on ∂R. The undeformed DVZ depends on the state vector π that characterizes the motion capabilities of the body (its translational and rotational velocities for instance):

$$\Xi_h = \beta(\pi) \tag{2}$$

Let M be a point of the boundary of R, $M \in \partial R$. The deformation vector $\triangle(M)$ depends on I, the intrusion of the environment into the body space at M and on the undeformed DVZ at M:

$$\triangle(M) = \alpha(\Xi_h(M), I(M)) \tag{3}$$

In the following equations, we refer to (3) as:

$$\triangle = \alpha(\Xi_h, I) \tag{4}$$

implicitly implying the vectors image of M through the correspondent map. Differentiating equation (4) yields:

$$\dot{\triangle} = \nabla_{\Xi_h}[\alpha]\nabla_{\pi}[\beta]\phi + \nabla_I[\alpha]\psi \tag{5}$$

where ∇_ξ is the derivation operator with respect to the variable ξ and where $\phi = \nabla_t \pi = \dot{\pi}$ and $\psi = \nabla_t I = \dot{I}$ are the two control vectors of $\dot{\triangle}$.

The evolution of $\triangle$ is driven by two-fold input vectors $u = [\phi, \psi]^T$. The first control vector ϕ, due to the controller, tends to minimize the deformation of the DVZ. The second one, ψ, is unknown and is induced by the environment (and could, at most, try to maximize the deformations). The complete evolution of the deformation is modeled by a differential equation of the type:

$$\dot{\triangle} = A\phi + B\psi \tag{6}$$

The DVZ control algorithm consists of choosing the desired evolution $\dot{\triangle}_{des}$ of the deformation and applying the following lemma [5]:

Lemma 1 (DVZ principle). *Given $\dot{\triangle}_{des}$, the best control vector $\breve{\phi}$ in the sense of least-squares that minimizes the function $\|\dot{\triangle}_{des} - \dot{\triangle}\|^2$ is obtained by inverting equation (6):*

$$\breve{\phi} = A^+(\dot{\triangle}_{des} - B\psi) \tag{7}$$

where A^+ is the pseudo-inverse of A.

A simple and efficient control law consist of choosing the desired deformation as proportional to the real deformation and its derivative:

$$\dot{\triangle}_{des} = -K_p\triangle - K_d\dot{\triangle} \tag{8}$$

where the two matrices K_p and K_d are respectively the proportional and derivative gains and are tuned in order to carry out the avoidance task. In this work

we define an ellipse as DVZ parameterized by the linear velocity and the steering angle of vehicle. Figure 1 shows different cases of the one-sphere deformations. These zones represent the various shapes of the DVZ, depending on the translational and rotational velocities of the robot. The first diagram illustrate a deformed DVZ due to the presence of an obstacle. The remaining diagrams show how the mobile robot can rebuild its DVZ, (b) by reducing the translational velocity, (c) by turning to the right, or (d) by turning to the left.

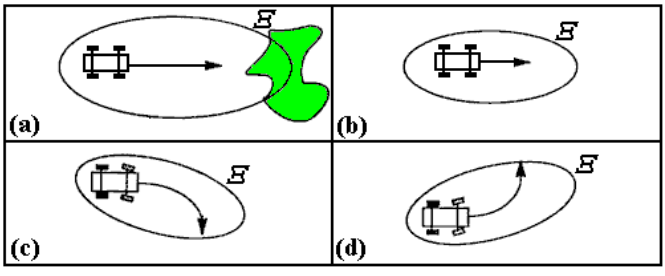


Fig. 1. Deformations of a 1-dimensional DVZ

4 A Reactive Lazy PRM Planner

The adaptation of PRM planners to environments with both static and moving obstacles has been limited so far. The proposed approach, integrates the lazy PRM planning method and the reactive control by DVZ in the following way: a collision-free feasible path for a mobile robot is calculated by the lazy PRM method, the robot starts moving (under the permanent protection of its DVZ), in the absence of dynamic obstacles, the control is performed by the lazy PRM method and does not require reflex commands. If there are dynamic obstacles in its path, the reactive method takes the control and generates commands to force the robot to move away from the intruder obstacles and gives back its DVZ to the original state. In this point, the robot has lost its original path, and it is necessary to search for a reconnection path to reach its goal. The new path found is a single collision-free curve of Reeds & Shepp [12]. If the attempt of reconnection is successful, the robot executes its new path towards the goal. The new alternative path was obtained with the lazy PRM method by using the information stored in the current robot's configuration, but if a deformation appears, the processes are interrupted by reflex actions that forces the planner to go back to the previous state. The algorithm can finish in three forms: i) the robot executes its path successfully, ii) the reflex action is not sufficient and a collision occurs, or iii) the robot does not find an alternative path to conclude its task. Figure 2 shows a high-level description of the proposed approach. The following subsections detail two of the most important stages of the proposed approach. For more details, you can see [13].

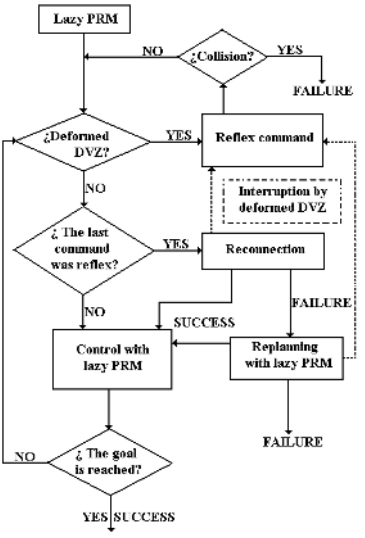


Fig. 2. High-level description of our proposed approach

4.1 Reconnection

After a successful reflex action, the mobile robot recovers the intact state of its DVZ, but its initial planned path will be lost (Fig. 3-b). The lazy PRM method needs to have a path to push the mobile robot to the goal and it will be necessary to provide a path for such aim. Due to the high computational cost of a complete replanning, the method will avoid it by executing a process that uses a single collision-free Reeds & Shepp curve (Fig. 3-c) to reconnect with the planned path. Initially, the algorithm tries a local path that it is interrupted by a dynamic object. The algorithm will execute a reflex action in order to reconnect with the closest point that is collision-free in the original path. If it can not reconnect after a certain number of attempts, maybe because the possible reconnection

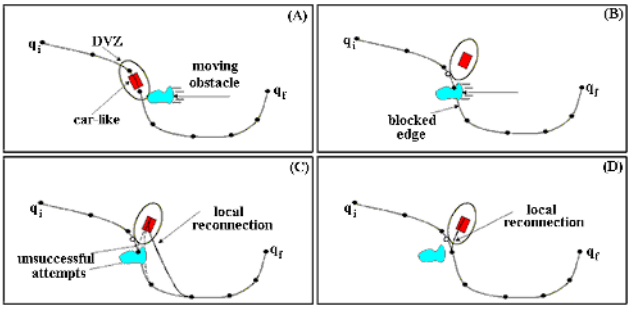


Fig. 3. Cases of the reconnection process: a) to avoid a dynamic obstacle, b) after a reflex action, c) after many previous attempts, c) a successful reconnection

paths are blocked with obstacles, the robot will remain immovable for a certain time before executing a new attempt (see Fig. 3-d). The process will be repeated several times, but if the DVZ was deformed by an intrusion, the reconnection process will be modified and will execute the reflex commands.

4.2 Replanning

If the reconnection attempts fails, it may happen that paths are blocked by many dynamic objects, or a moving object is parked obstructing the planned path. In this case, the planner executes the lazy PRM method (the initial configuration is the current configuration for the robot). The lazy PRM will be called several times until it returns a collision-free path. If after some attempts a collision-free path can not be found, the planner reports failure. The model cannot distinguish if an intrusion is caused by a moving or a static obstacle because the DVZ method does not use any model of the environment. To solve this problem, it is necessary to use an auxiliary image that represents the environment and it is updated every time the replanning or reconnection procedures are called. When the sensors in the robot detect an obstacle that deforms the DVZ, the intruder object coordinates are revised to see if there was already an obstacle, registered in the auxiliary image; if this is the case, the system assumes the presence of a fixed obstacle and there is no need for a reflex action, otherwise, it will certainly assume that the object is in movement.

5 Experimental Results

This section presents experimental results for car-like robots. The planner has been implemented in Builder C++ and the tests were performed on an Intel © Pentium IV processor-based PC running at 2.4 GHz with 512 MB RAM. The moving obstacles have a square form and move at constant velocity in straight line. Whenever they collide with another object they assume a new random direction in their movement. Fig. 4 shows an environment composed of four static obstacles and several dynamic obstacles moving randomly at the same velocity than the mobile robot.

In order to evaluate the performance of the planner, we performed tests on the environment of Fig. 5 for several roadmap sizes and different number of moving obstacles. The different settings are summarized in the tables 1 and 2. In our case, due to the strategy of node addition, the time for the roadmap's construction is proportional to the number of nodes. The number of nodes at the beginning is a critical parameter that affects the lazy PRM's performance [10], [9]. To show the methodology proposed, we performed 50 trials.

In fact, the method's performance can be considered satisfactory if it presents a fast planning phase, reflex actions based on sensors that do not require expensive algorithms, an effective process of reconnection performed in milliseconds, and a process of replanning that is executed if the Lazy PRM and DVZ' s parameters are appropriate. As mentioned in earlier sections, it can be considered

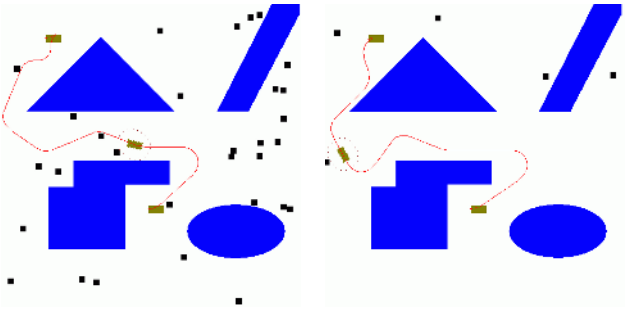


Fig. 4. An example of a query and its solution path (left) in an environment with 30 moving obstacles. The robot starts moving under the permanent protection of its DVZ (right), the scene contains 5 moving obstacles.

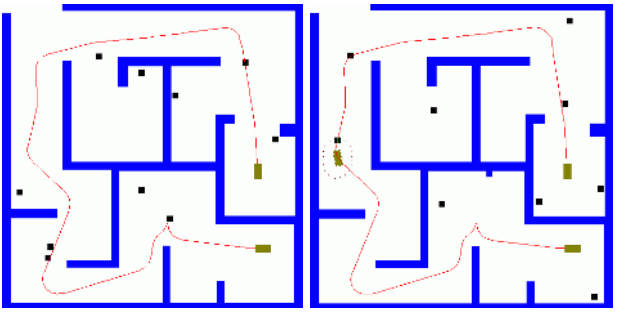


Fig. 5. An environment composed of some narrow passages. The scene contains 10 moving obstacles.

that the methodology proposed here, includes these characteristics. The planning time is reduced due to the incomplete collision detector whose work is complemented with the robot's sensors during the path execution. On the other hand, the assignation of direction angles to the nodes that conform the shortest paths obtained by the algorithm A^*, produces curves that allow the algorithm to omit the optimization process (i.e., the smoothing process). With respect to the reconnection process, the paths obtained with the planner are conformed

Table 1. Performance data for Lazy PRM

Settings	50 nodes	100 nodes	50 nodes	100 nodes	200 nodes	400 nodes
Steering angle	45	45	30	30	45	45
Graph building	0.007	0.017	0.009	0.028	0.054	0.145
Graph searching	0.277	0.618	0.185	0.190	1.404	2.453
Coll. checking	11785	9539	18466	11383	10168	7269
Total Time (s)	1.843	2.058	1.623	0.963	2.952	3.600

by a single Reeds & Shepp curve and based on the incomplete collision detector, making short the time and close to optimal the curves obtained with the algorithm. Since the reflex actions are provided by the DVZ method, it is possible to interrupt the reconnection and replanning processes if necessary, without incurring in bigger problems.

Table 2. Performance data with 10 moving obstacles

Reconnections	Time for reconnection	Replanning	Time for replanning	Collision	Success
50	0.035	0	0	ok	ok
247	0.040	2	1.481	no	ok
209	0.034	1	0.427	no	ok
260	0.042	3	0.639	no	ok
93	0.051	1	1.088	ok	ok
59	0.041	1	2.445	no	ok

6 Conclusions and Future Work

Even in the absence of obstacles, planning motions for nonholonomic systems is not an easy task. So far, no general algorithm exists for planning the motions of any nonholonomic system, that guarantees to reach a given goal. The only existing results deal with approximation methods, that is, methods that guarantees to reach a neighborhood of the goal, and exact methods for special classes of nonholonomic systems. Obstacle avoidance adds a second level of difficulty: not only does one have to take into account the constraints imposed by the kinematic nature of the system, but also the constraints due to the obstacles. It appears necessary to combine geometric techniques addressing the obstacle avoidance with control techniques addressing nonholonomic motions.

A reactive lazy PRM planner for dynamically changing environments is presented in this paper. The results obtained in the evaluation of the reactive lazy PRM planner proposed in this work, show the importance of finding a solution for the complex problem of motion planning for nonholonomic robots in dynamic environments. Although some promising results are shown in its present form, the planner can be improved in a number of important ways. This approach can be extended to use real robots and to solve the problem posed by small static obstacles. Besides, some cases where the reflex actions are not sufficient to avoid collisions, were observed during the evaluation tests. Theses cases are difficult because they require a more intelligent behavior in order to avoid the robot to be trapped. In those cases, it can be necessary to add a process that computes the trajectories of moving objects and corrects the robot's path in real time.

Finally, this methodology can be extended to study non-structured environments, solving one of the more interesting research topics in robotics.

References

1. Brooks R. A. "Robot beings", *International Workshop on Intelligent Robots and Systems*, (1989) 2-10
2. Anderson T. L. "Autonomous robots and emergent behaviours: A set of primitive behaviours for mobile robot control", *IROS 90*, (1990)
3. O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots", *International Journal of Robotic Research*, Vol. 5, No. 1, (1986) 90-98
4. Zapata R. "Quelques aspects topologiques de la planification de mouvements et des actions réflexes en robotique mobile", *Thèse d'Etat*, University of Montpellier II, (1991)
5. Zapata R., Lépinay P., and Thompson P. "Reactive behaviors of fast mobile robots", *Journal of Robotic Systems*, Vol. 11, No. 1, (1994) 13-20
6. Kavraki L. E., Švetska P., Latombe J. C., and Overmars M. H. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, (1996) 566-580
7. Švestka P. and Overmars M. H. "Motion planning for car-like using a probabilistic learning approach", *The International Journal of Robotics Research*, Vol. 16, No. 2, (1997) 119-143
8. Hsu D., Kindel R., Latombe J. C., and Rock S. "Randomized kinodynamic motion planning with moving obstacles", *Workshop on Algorithmic Foundations on Robotics* (2000)
9. Bohlin R. and Kavraki L. E. "Path planning using lazy PRM", *Proc. of the IEEE Robotics and Automation Conference*, (2000) 521-528
10. Sánchez A., Zapata R., Arenas A. "Motion planning for car-like robots using a lazy probabilistic roadmap method", *Advances in Artificial Intelligence, Springer Verlag, Vol. 2313*, (2002) 1-10
11. Sánchez A., Zapata R., and Lanzoni C. "On the use of low-discrepancy sequences in nonholonomic motion planning", *Proc. of the IEEE Robotics and Automation Conference*, (2003) 3764-3769
12. Reeds J. A. and Shepp R. A. "Optimal paths for a car that goes both forward and backwards", *Pacific Journal of Mathematics*, Vol. 145, No. 2, (1990) 367-393
13. Cuautle P. R. "Planificación de movimientos en ambientes dinámicos utilizando un enfoque lazy PRM reactivo", *Tesis de Maestria, FCC-BUAP*, (in spanish), 2005.

Negative Information in Cooperative Multirobot Localization*

Valguima Odakura and Anna Helena Reali Costa

Laboratório de Técnicas Inteligentes - LTI
Depart. de Eng. de Computação e Sistemas Digitais - PCS
Escola Politécnica da Universidade de São Paulo - EPUSP
{valguima.odakura, anna.reali}@poli.usp.br

Abstract. This paper proposes the use of negative detection information to improve multirobot Markov localization. In multirobot localization, the pose beliefs of two robots are updated whenever one robot detects another one and measures their relative distance. Negative detection information means absence of detection information and in general is not used in the updating of pose beliefs. However, it can also be informative and we argue that this information should be incorporated into the localization approach. We contribute to a negative detection model and show how it can be integrated into cooperative multirobot Markov localization. Experimental results show that the use of negative detection information leads to an improvement of localization accuracy.

1 Introduction

Localization is one of the main abilities of an autonomous mobile robot. In order to perform their tasks, mobile robots need to know their poses within the environment. To accomplish this task robots use their sensor measurements which provide information about robot's movement and about the environment. A very known localization approach is called Markov localization [1], that incorporates the sensor measurements based on two models, a motion model and an observation model.

In the multirobot localization each robot can use measurements taken by all robots, in order to better estimate its own pose. In this way, the main difference between single robot and cooperative multiple robots localization is that multirobot can achieve more information than a single robot. The Markov localization approach incorporates multirobot measurements by using detection information to update robots' poses, it means that when a robot detects and identifies another robot, they can exchange their pose beliefs in order to get more information for self-localization task.

All sensor information provided to a single or multirobot in the Markov localization approach are positive information in the sense that it represents a

* This work was supported by Fapesp grant 01/14246-4, Capes grant BEX 0803/03-0, CNPq and Capes/Grices 099/03.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 552–561, 2006.

sensor measurement of important features of the environment. In this paper we are interested in the case where the sensor information acquired represents that no important features were detected – **the negative information**.

Human beings often use negative information. For example, if you are looking for someone in a house, and you do not see the person in a room, you can use this negative information as an evidence that the person is not in that room, so you should look for him/her in another place.

The aim of this paper is to show how negative information can be incorporated in multirobot localization. The idea is that even a failed attempt to detect a robot is a useful information, which can be used to improve localization accuracy.

This paper is organized as follows. In Section 2, the single robot localization problem is presented and the Markov localization technique is introduced. Section 3 extends the localization problem to a group of robots. The proposed negative detection model to improve multirobot localization accuracy is described in Section 4. Experimented results are shown in Section 5. Finally, in Section 6, our conclusions are derived and future works are presented.

2 Robot Localization

Mobile robot localization is the problem of estimating a robot's pose within an environment based on observations. Observations consist in information about the robot's movement and about the environment. Information provided by sensors are inherently uncertain, so probabilistic techniques are needed to deal with this.

A very known localization technique is Markov localization – ML. This localization technique maintains a probability distribution over the space of all poses of a robot in its environment, so it deals with multimodal distributions. Markov localization relies on the Markov assumption, which states that future sensor readings are conditionally independent of past readings, given the true pose of the robot [1].

In ML, $p(\mathbf{x}_t = x)$ denotes the robot's belief that it is at pose x at time t, where $\mathbf{x}_t$ is a random variable representing the robot's pose at time t, and $x = (\mathsf{x}, \mathsf{y}, \theta)$ is the pose of the robot. This belief represents a probability distribution over all the space of $\mathbf{x}_t$.

ML uses two models to localize a robot: a motion model and an observation model. The motion model is specified as a probability distribution $p(\mathbf{x}_t = x|\mathbf{x}_{t-1} = x', \mathbf{a}_{t-1})$, where $\mathbf{x}_t$ is a random variable representing the robot's pose at time t, $\mathbf{a}_t$ is the action or movement executed by the robot at time t. The movement can be estimated, for example, by odometers on the wheels. In ML the update equation is described as:

$$p(\bar{\mathbf{x}}_t = x) = \sum_{x'} p(\mathbf{x}_t = x|\mathbf{x}_{t-1} = x', \mathbf{a}_{t-1}) p(\mathbf{x}_{t-1} = x'), \tag{1}$$

where $p(\bar{\mathbf{x}}_t = x)$ is the probability density function before incorporating the observation of the environment at time t.

The observation model is used to incorporate information from exteroceptive sensors, such as proximity sensors and camera, and it is expressed as $p(\mathbf{x}_t = x|\mathbf{o}_t)$, where $\mathbf{o}_t$ is an observation sensed at time t. The update equation is described as:

$$p(\mathbf{x}_t = x) = \frac{p(\mathbf{o}_t|\mathbf{x}_t = x)p(\bar{\mathbf{x}}_t = x)}{\sum_{x'} p(\mathbf{o}_t|\mathbf{x}_t = x')p(\bar{\mathbf{x}}_t = x')}, \tag{2}$$

where $p(\mathbf{x}_t = x)$ is the probability density function after incorporating the observation of the environment at time t.

In the beginning, $p(\mathbf{x}_0 = x)$ is the prior belief about the initial pose of the robot. If the initial pose is unknown, $p(\mathbf{x}_0 = x)$ is uniformly distributed over all possible poses.

3 Cooperative Multirobot Localization

Multiple robots exhibiting cooperative behavior is interesting for two main reasons: it may be easier and cheaper to use several simple cooperative robots than a single powerful one to accomplish a difficult task, and a group of robots could cover a big area faster than a single one [2]. In order to accomplish tasks more quickly and robustly than a single robot, multiple robots can be used, which poses the multirobot localization problem.

The cooperative multirobot localization problem consists in localizing each robot in a group within the same environment, when robots share information in order to improve localization accuracy.

Representative recent works in cooperative multirobot localization are from Roumeliotis and Bekey [3] and Fox et al. [4], that use Kalman filter and Particle filter as localization algorithms, respectively.

Markov localization was initially designed for a single robot [1]. An extension of ML that aims at solving the multirobot problem is presented by Fox et al. [4]. In order to accommodate multirobot cooperation in ML it is necessary to add a detection model to the previous observation and motion models.

The detection model [4] is based on the assumption that each robot is able to detect and identify other robots and furthermore, the robots can communicate their probabilities distributions to other robots. Let's suppose that robot n detects robot m and measures the relative distance between them, so:

$$p(\mathbf{x}_t^n = x) = p(\mathbf{x}_{t-1}^n = x) \sum_{x'} p(\mathbf{x}_{t-1}^n = x|\mathbf{x}_{t-1}^m = x', \mathbf{r}_n)p(\mathbf{x}_{t-1}^m = x'), \tag{3}$$

where $\mathbf{x}_t^n$ represents the pose of robot n, $\mathbf{x}_t^m$ represents the pose of robot m and $\mathbf{r}_n$ denotes the measured distance between robots. The calculation $\sum_{x'} p(\mathbf{x}_{t-1}^n = x|\mathbf{x}_{t-1}^m = x', \mathbf{r}_n)p(\mathbf{x}_{t-1}^m = x')$ describes the belief of robot m about the pose of robot n. Similarly, the same detection is used to update the pose of robot m.

To summarize, in the multirobot localization algorithm each robot updates its pose belief whenever a new information is available in the following Equation sequence: (1) is used when a motion information is available; (2) is used when

an environment observation occurs; and when a detection happens, both robots involved use (3) to update their beliefs of poses.

Once a detection is made according to the detection model, the two robots involved in the process share their probabilities distributions and relative distance. This communication significantly improves the localization accuracy, if compared to a less communicative localization approach.

One disadvantage of this approach is that detection information is shared only by the two meeting robots and it is not used by the other robots in the group. In a previous work we have presented a detection model where all robots of the group can benefit from a meeting of two robots through the propagation of the meeting information [5].

4 Negative Information

Negative information measurement means that at a given time, the sensor is expected to report a measurement but it did not. Most of the techniques of state estimation use a sensor model that update the state belief when the sensor reports a measurement. However it is possible to get useful information of the state from the absence of sensor measurements relative to the expected measurement landmarks.

When a robot knows the map of the environment and has a sensor model it can use sensor measurements of a landmark to calculate its pose in the environment. This is the positive information used in the ML approach. There are two main reasons for robot's sensors not to measure a landmark. The first one is that the expected landmark is not in the field of view of the robot and, the second one is that the sensor is unable to detect the landmark, due to occlusions, sensor errors, etc.

This situation of non detecting a landmark can be modeled by considering the robot field of view and by using an obstacle detection to identify occlusions, as shown:

$$p(\mathbf{x}_t = x) = \frac{p(\mathbf{o}_t^- | \mathbf{x}_t = x, \mathbf{v}, \mathbf{obs}_t) p(\bar{\mathbf{x}}_t = x)}{\sum_{x'} p(\mathbf{o}_t^- | \mathbf{x}_t = x, \mathbf{v}, \mathbf{obs}_t) p(\bar{\mathbf{x}}_t = x')}, \tag{4}$$

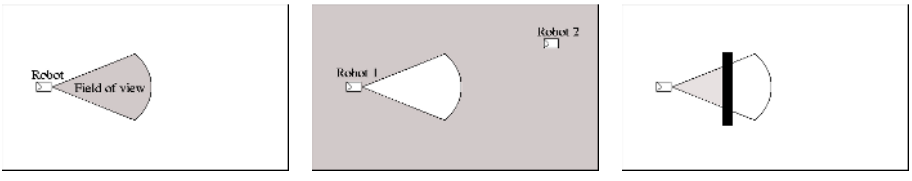
where $\mathbf{o}_t^-$ represents the negative information, $\mathbf{v}$ describes the visibility area of the sensor and $\mathbf{obs}_t$ represents the occlusion area.

The negative information has been applied to target tracking using the event of not detecting a target as evidence to update the probability density function [6]. In that work a negative information means that the target is not located in the visibility area of the sensor and since the target is known to exist it is certainly outside this area.

In robot localization domain, the work of Hoffmann et al. [7] on negative information in ML considers as negative information the absence of landmark sensor measurements. Occlusions are identified using a visual sensor that scans colors of the ground to determine if there is free area or obstacle. The environment is a soccer field in green with white lines. So, if a different color is identified, it means that an obstacle could be occluding the visibility of a landmark.

In the cooperative multirobot localization problem negative information can also mean the absence of detections (in the case that a robot does not detect another one), which configures a lack of group information. In this case, the negative detection measurement can provide the useful information that a robot is not located in the visibility area of another robot. In some cases, it can be an essential information as it could improve the pose belief of a robot that is located at a part of environment with few discriminant landmarks.

Our contribution in this paper is the proposal of a negative detection model and its incorporation into multirobot ML approach. Consider two robots within a known environment and their field of view as shown in Fig. 1(a). If robot 1 does not detect robot 2 at a given point in time, a negative information is reported, which states that robot 2 is not in the visibility area of robot 1, as depicted in Fig. 1(b).



(a) (b) (c)

Fig. 1. (a) Robot field of view. (b) Negative information. (c) Occlusion in the field of view.

The information gathered from Fig. 1(b) is true if we consider that there are no occlusions. In order to account for occlusions it is necessary to sense the environment to identify free areas or occupied areas. If there is a free space on the visibility area of a detection sensor, than there is not an occlusion. Otherwise, if it is identified as an occupied area it means that the other robot could be occluded by another robot or an obstacle. In this case it is possible to use geometric inference to determine which part of the visibility area can be used as negative detection information, as shown in Fig. 1(c).

Let's suppose that robot m makes a negative detection. The negative detection model, considering the visibility area of the robot and the occlusion area, becomes:

$$p(\mathbf{x}_t^{m-} = x) = \frac{p(\mathbf{d}_t^- | \mathbf{x}_t^m = x, \mathbf{v}, \mathbf{obs}) p(\mathbf{x}_t^m = x)}{\sum_{x'} p(\mathbf{d}_t^- | \mathbf{x}_t^m = x', \mathbf{v}, \mathbf{obs}) p(\mathbf{x}_t^m = x')}, \tag{5}$$

where $\mathbf{d}_t^-$ is the event of not detecting any robot and $\mathbf{x}_m$ corresponds to the state of robot m, the robot that reports the negative detection information. The variables $\mathbf{v}$ and $\mathbf{obs}$ represent the visibility area and the identified obstacles, respectively.

Whenever a robot m does not detect another robot k, we can update the probability distribution function of each k, with $k \neq m$, in the following way:

$$p(\mathbf{x}_t^k = x) = \frac{p(\bar{\mathbf{x}}_t^k = x)p(\mathbf{x}_t^{m-} = x)}{\sum_{x'} p(\bar{\mathbf{x}}_t^k = x')p(\mathbf{x}_t^{m-} = x')}, \tag{6}$$

where $\mathbf{x}_k$, for $k = 0, \cdots, n$, represents all robots which were not detected.

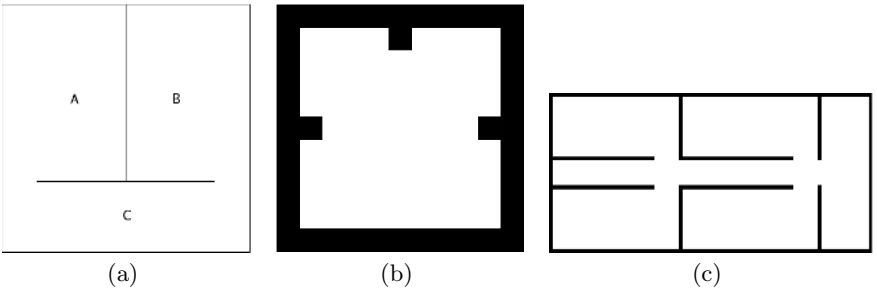
In the case that a positive detection is reported, the robots involved in the detection update their beliefs according to (3).

Negative detection model allows solving certain localization problems that are unsolvable for a group of robots that only relies on positive detection information. A typical situation is the case of robots in different rooms, in a way that one robot cannot detect the other.

5 Experiments

In order to evaluate the localization results obtained with the negative detection model proposed in this paper, we perform some experiments. The experiments are conducted with simulated robots. Each robot is equipped with a proximity sensor to measure the distance to the walls in the environment, and a detection sensor, that can identify other robots and measure their relative distance. All robot's sensors are assumed to be corrupted by Gaussian noise. The robots know an environment model and they do not know their initial poses in the environment.

The test environment, shown in Fig. 2(a), has dimensions of 3×3 meters, and is a symmetric environment, with two rooms A and B and a corridor C. The robot environment model is based on a grid, where each cell has dimensions of 0.2×0.2 meters, and angular resolution of 90 degrees. It results in a state space of dimension $15 \times 15 \times 4 = 900$ states.



(a) (b) (c)

Fig. 2. Test environment: (a) Two symmetric rooms (A and B) and a corridor (C). (b) Open environment. (c) Rooms and corridor.

We conducted a preliminary experiment with two robots. Robot 1 is located in room A and robot 2 in room B. At a given moment, robot 1 knows accurately

its pose, and robot 2 is in doubt about being in room A or room B. Due to the environment symmetry it is impossible to robot 2 to find out that its real pose is in room B and since the robots are located at different rooms, one robot can not detect each other. However, robot 2 would be able to localize itself if negative detection information could be used to update its pose belief. Fig. 3 depicts this situation. Fig. 3(a) and Fig. 3(b) show the probability density function of robot 1 and robot 2, respectively. In Fig. 3(c) is shown the negative detection information derived from the belief of robot 1 and its visibility area. When robot 2 updates its pose using the negative detection information reported by robot 1, it becomes certain about its pose, as shown in Fig. 3(d).

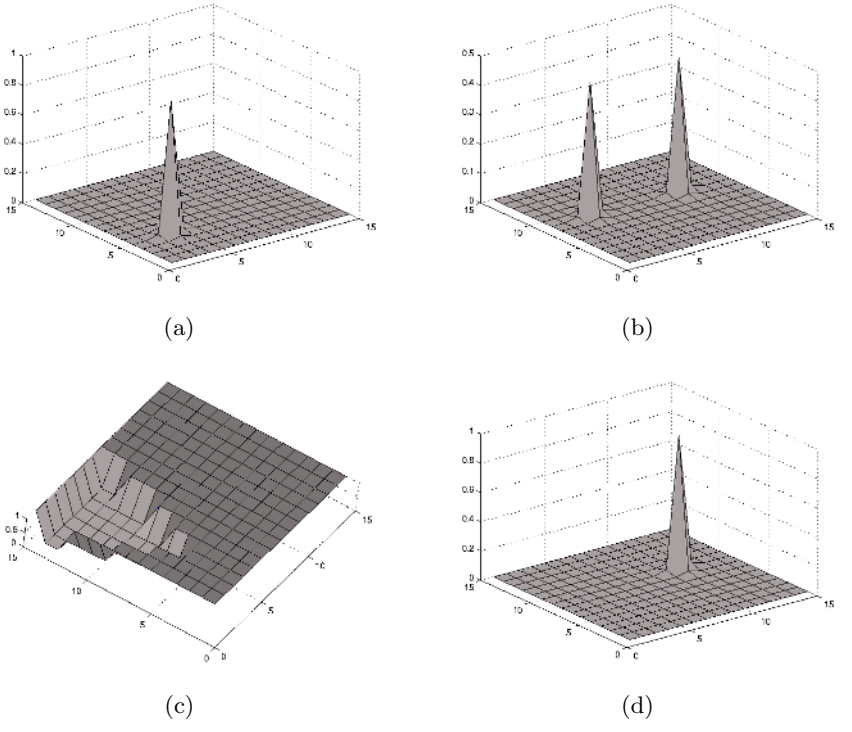


Fig. 3. (a) Probability density function of robot 1. (b) Probability density function of robot 2. (c) Negative detection information reported by robot 1. (d) Probability density function of robot 2 after the incorporation of negative detection information.

The following experiment is a localization task using two robots. The robots move in the environment and collect information in order to improve their localization beliefs. In one run the robots perform the multirobot ML and in the next one, robots perform multirobot ML with negative detection information.

In order to compare the experimental results obtained using or not the negative detection information, we use the expected entropy value H as a quality

measure of the pose estimation. From information theory [8], the entropy measures the amount of information provided by the probability density function, and is defined as:

$$H = -\sum_{x} P(\mathbf{x}_t = x) \log(P(\mathbf{x}_t = x). \tag{7}$$

The entropy value H is zero if the robot is perfectly localized in one pose, and the entropy value achieves maximal value if the robot is uncertain about its pose, which means that the probability density function is uniformly distributed.

In this experiment robot 1 starts being well localized and robot 2 is completely lost. Fig. 4(a) shows the entropy values for both runs for robot 2. The solid line represents entropy results obtained with positive information and the dashed line represents entropy results obtained with negative detection information. At first the robot does not know its pose and the expected entropy has the maximal value. As soon as the robot updates its pose with the information reported from the environment the entropy value drops. Using negative detection information, the accuracy of the localization is greatly improved.

The next experiment is executed on the same environment with three robots. We perform four runs, where each run presents a different initial pose for each robot. The result is shown in Fig. 4(b).

To explore a different environment, we test our approach in an open environment as depicted in Fig. 2(b). The environment has dimensions of 4×4 meters. The robot environment model is based on a grid, where each cell has dimensions of 0.2×0.2 meters, and angular resolution of 90 degrees. The results of the experiment with four robots, averaged over six runs, is shown in Fig. 5(a). The increase in localization accuracy is not significant, as the environment is an open area without ambiguities, and as so, does not present the situation where the negative detection information is crucial.

Similar tests were conducted with four robots in the environment shown in Fig. 2(c). The environment has dimensions of 9×4 meters and is a standard office environment with some rooms and a corridor. The result of the experiment is depicted in Fig. 5(b), where the use of negative detection information contributes to improve the localization accuracy.

It is worth mentioning the amount of information communicated with the use of the negative detection model, which is a probability density function propagated to all other robots of the group. The number of robots involved in the communication can be reduced as the robots become more certain about their poses, which means that a robot that knows its pose does not need to use information from another robot to improve its own pose belief. There is a tradeoff between robot communication and localization accuracy, that we plan to investigate in future work.

Localization results obtained with negative detection information into cooperative multirobot Markov localization are more accurate and provide the ability to localize robots more quickly. These are highly beneficial in real world applications where robots need to actually perform a task rather than to localize perfectly.

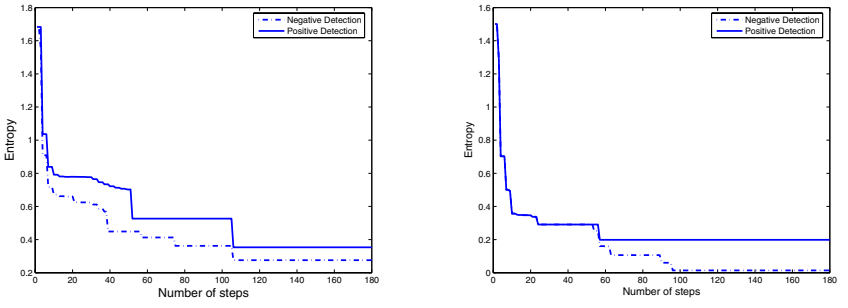


Fig. 4. Expected entropy values of the belief in localization with and without negative detection for environment of Fig. 2(a): (a) Two robots. (b) Three robots.

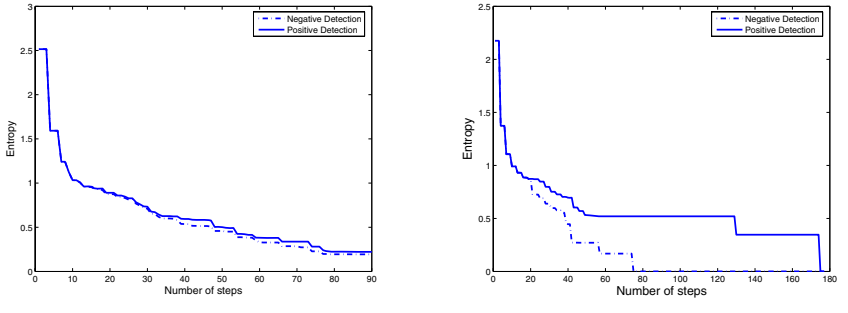


Fig. 5. Expected entropy values of the belief in localization with and without negative detection: (a) Four robots, environment of Fig. 2(b). (b) Four robots, environment of Fig. 2(c).

6 Conclusions

In this paper we propose a negative detection model and show how it can be integrated into the cooperative multirobot ML.

Experimental results demonstrate that our approach, when compared to previous multirobot localization methods without negative detection, reduce the uncertainty in the localization significantly.

We have shown that using negative detection information, a robot is able to localize itself in environments where otherwise it would find it more difficult to do it. In this way, we give a contribution in the direction of a precise localization, that is one of the main requirements for mobile robot autonomy.

Finally, as part of our future work, we plan to investigate other forms of information to update robot's pose belief. We also intend to explore the tradeoff between the amount of information communicated among robots and the localization accuracy.

References

1. Fox, D., Burgard, W., Thrun, S.: Markov localization for mobile robots in dynamic environments. Journal of Artificial Intelligence Research **11** (1999) 391–427
2. Cao, Y.U., Fukunaga, A.S., Kahng, A.B., Meng, F.: Cooperative mobile robotics: antecedents and directions. Autonomous Robots **4** (1997) 1–23
3. Roumeliotis, S.I., Bekey, G.A.: Distributed multirobot localization. IEEE Transactions on Robotics and Automation **18**(2) (2002) 781–795
4. Fox, D., Burgard, W., Kruppa, H., Thrun, S.: A probabilistic approach to collaborative multi-robot localization. Autonomous Robots **8**(3) (2000)
5. Odakura, V., Costa, A.H.R.: Cooperative multi-robot localization: using communication to reduce localization error. In: International Conference on Informatics in Control, Automation and Robotics - ICINCO'05. (2005) 88–93
6. Koch, W.: On negative information in tracking and sensor data fusion: discussion of selected examples. In: Seventh International Conference on Information Fusion. (2004) 91–98
7. Hoffmann, J., Spranger, M., Gohring, D., Jungel, M.: Making use of what you dont see: Negative information in markov localization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS'05, IEEE (2005) 854–859
8. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall (1995)

Gait Control Generation for Physically Based Simulated Robots Using Genetic Algorithms

Milton Roberto Heinen and Fernando Santos Osório

Unisinos - PIPCA - Applied Computing
São Leopoldo, CEP 93022-000, Brazil
mheinen@turing.unisinos.br, fosorio@unisinos.br

Abstract. This paper describes our studies in the legged robots research area and the development of the LegGen System, that is used to automatically create and control stable gaits for legged robots into a physically based simulation environment. The parameters used to control the robot are optimized using Genetic Algorithms (GA). Comparisons between different fitness functions were accomplished, indicating how to compose a better multi-criterion fitness function to be used in the gait control of the legged robots. The best gait control solution and the best robot model were selected in order to help us to build a real robot in the future. The results also showed that it is possible to generate stable gaits using GA in an efficient manner.

1 Introduction

The autonomous mobile robots has been attracting the attention of a great number of researchers, due to the challenge that this new research domain proposes: make these systems capable of intelligent reasoning and able to interact with the environment they are inserted in, through sensor's perception (infrared, sonar, bumpers, gyroscopes, etc) and motor's action planning and execution[1]. At the present time, the most part of mobile robots use wheels for locomotion, what does this task easy to control and efficient in terms of energy consumption, but they have some disadvantages since they have problems to move across irregular surfaces and to cross borders and edges. So, in order to make mobile robots better adapted to human environments and to irregular surfaces, they must be able to walk and/or to have a similar locomotion mechanism used by the humans and animals, that is, they should have a legged locomotion mechanism[1].

However, the development of legged robots capable to move in irregular surfaces is a quite difficult task, that needs the configuration of many gait parameters[2]. The manual configuration of these parameters demands a lot of effort and spent time of a human specialist, and the obtained results are usually suboptimal and specific for one robot architecture[3]. Thus, it is interesting to generate the robot gait configuration in an automatic manner, using Machine Learning techniques to perform this task.

One of these Machine Learning techniques that are most adapted for this specific task are the Genetic Algorithms (GA)[4,5]. This is a reasonable choice

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 562–571, 2006.

because according to the Evolution's Theory[6], the locomotion mechanisms of several life forms resulted from the natural evolution, what makes the use of Genetic Algorithms a natural solution since they are biologically inspired and can generate biologically plausible solutions. From the computational point of view, the Genetic Algorithms are also very well adapted for the automatic gait configuration of legged robots, because: (a) they use a multi-criterion optimization method to search solutions in the configuration space, that means in our specific case, they are capable to optimize not only the gait velocity, but also the stability and even other gait parameters; (b) they don't need local information for the error minimization, nor the gradient calculation, what is very important for the gait parameters generation and optimization, since it is very difficult to have available some a priori training data for supervised learning; (c) if correctly used, the Genetic Algorithms are capable to avoid local minima[5].

The main goal of this paper is to describe the LegGen System[7], that is capable to automatically evolve the gait control of physically based simulated legged robots using Genetic Algorithms. This paper is structured as follows: The Section 2 describes the Genetic Algorithms and the GAlib software library adopted in our system; The Section 3 describes several concepts relative to legged robots, as the static and dynamic stability, the use of a physical simulation engine, the legged robot configuration, and the fitness functions used in the experiments; The Section 4 describes the LegGen system, and the robots used in the simulations; The Section 5 describes the executed experiments and the obtained results; and the Section 6 provides some final conclusions and future perspectives of this work.

2 Genetic Algorithms

Genetic Algorithms are optimization methods of stochastic space state search based on the Darwin's Natural Evolution Theory[6], that were proposed in the 60s by John Holland[8]. They work with a population of initial solutions, called chromosomes, which are evolved through several operations during a certain number of generations, usually finding a sub-optimal solution, and preserving the best individuals according to a specific evaluation criterion. In order to accomplish this, in each generation the chromosomes are individually evaluated using a function that measures its performance, called fitness function[5]. Individuals are selected to generate the next generation with probability proportional to their fitness values, and the crossover and mutation operations are applied in these individuals. Thus, each new generation tends to adapt and improve the quality of solutions, until we obtain a solution that satisfies a specific objective[4].

The Genetic Algorithms implementation used in our system was based on the GAlib software library[1], developed by Matthew Wall of Massachusetts Institute of Technology (MIT). GAlib was selected as it is one of the most complete, efficient and well known libraries for Genetic Algorithms simulation, and also it is a free open source library based on C++. In the LegGen System, a simple

[1] GAlib – http://www.lancet.mit.edu/ga/

Genetic Algorithm was used with a floating point type genome. In order to reduce the search space, alleles were used to limit generated values only to possible values for each parameter (maximum and minimum joint angles). In the executed experiments (described in Section 5) uniform crossover with probability of 0.6 was used, Gaussian mutation with probability of 0.05, population size of 50 individuals and 100 maximum generations.

3 Legged Robots

In this section, important concepts related to gait generation and control for legged robots are described, beginning with the stability concept.

3.1 Stability

A very important concept related to legged robot's gait is the stability. Thus, in order to make the robot move into an environment avoiding to fall down, it is necessary to have a stable gait, and this stability can be static or dynamic[1]. A robot is said to exhibit static stability when the robot's center of gravity remains inside the convex hull of the support polygon defined by the legs currently touching the ground. The major advantage of static stability is that the robot do not risk to fall down if it remains static during a certain period of time, where this stability can be maintained while an other leg moves or even if an energy failure occurs.

If the center of gravity of the robot is allowed to move outside of the support polygon convex hull and the robot continues to move in a controlled manner, the robot is said to exhibit dynamic stability. The dynamic stability is more difficult to reach, because it demands a sophisticated model of robot's dynamics and the use of inertia[1].

3.2 Mobile Robot Simulation

When someone wants to make experiments in the mobile robots research area, two alternatives are possible: (a) to execute the experiments directly in a real robot; or (b) to make experiments using a simulated robot. The use of a real robot has the advantage of be realistic, but the simulation have the following advantages:

- When using simulated robots, doesn't exist the risk of robot damages and tasks as the exchange and recharge of batteries are not necessary[9];
- The robot positioning in order to restart a simulation can be accomplished automatically, without human intervention;
- The simulation clock can be accelerated, reducing the total amount of spent time for learning.

For these reasons, we chose to implement our initial experiments using a simulated robot, through the implementation of a very realistic robot simulator, using a physical simulation engine, so we can build simulated robots very similar to the real models.

3.3 Physics Simulation Engine

In order to do more realistic mobile robots simulation, several elements of the real world should be present in the simulated model, making the simulated bodies to behave in a similar way related to the reality. Especially, it is necessary that the robot suffers from instability and falls down if badly positioned and controlled, and also it should interact and collide against the environment objects in a realistic manner. To accomplish that, it is necessary to model the physics laws in the simulation environment (e.g. gravity, inertia, friction, collision). Nowadays, several physics simulation tools exist used for the implementation of physics laws in simulations. After study different possibilities, we chose a widely adopted free open source physics simulation library, called Open Dynamics Engine - ODE[2].

ODE is a software library for the simulation of articulated rigid bodies dynamics. With this software library, it's possible to make autonomous mobile and legged robots simulations with great physical realism. In ODE, several rigid bodies can be created and connected through different types of joints. To move bodies using ODE, it's possible to apply forces or torques directly to the body, or it is possible to activate and control angular motors. An angular motor is a simulation element that can be connected to two articulated bodies, which have several control parameters like axis, angular velocity and maximum force. With these elements, it's possible to reproduce articulations present in real robots, humans or animals, with a high precision level.

3.4 Gait Generation

In the LegGen System the gait control is generated using a Finite State Machine (FSM), in which is defined for each state and for each robot joint their final expected angles configuration[7]. In this way, the controller needs to continually read the joints angle state, in order to check if the joint motor accomplished the task. Real robots do this using sensors (encoders) to control the actual angle attained by the joints[1]. So, in this approach the gait control is accomplished in the following way: initially the controller verify if the joints have already reached the expected angles. The joints that do not have reached them are moved (activate motors), and when all the joints have reached their respective angles, the FSM passes to the following state. If some of the joints have not reached the specified angles after a certain limited time, the state is advanced independently of this. In a future version of the system, we are planning to treat this situation more carefully, because the leg can be blocked by an obstacle and the robot can be damaged in this case.

To synchronize the movements, it is important that all joints can reach their respective angles at almost the same time. This is possible with the application of a specific joint angular velocity for each joint, calculated by the equation:

$$V_{ij} = Vr_i(\alpha_{ij} - \alpha_{ij-1}), \tag{1}$$

[2] ODE – http://www.ode.org

where V_{ij} is the velocity applied to the motor joint i in the j state, α_{ij} is the joint angle i in the j state, α_{ij-1} is the joint angle i in $j-1$ state, and Vr_i is the reference velocity of the i state, used to control the set velocity. The reference velocity Vr is one parameter of the gait control that is also optimized by the Genetic Algorithm. The other parameters are the joints angles for each state. To reduce the search space, the Genetic Algorithm only generates values between the maximum and minimum accepted values for each specific parameter.

3.5 Fitness Function

In this work, different fitness functions were studied and tested to evaluate their contribution in order to generate a better gait control. The fitness function F of the Genetic Algorithm used in the first set of experiments was based only in the distance covered by the robot D in the x axis:

$$F = D = (Px_1 - Px_0)\,, \tag{2}$$

where Px_0 is the robot start position and Px_1 is the final robot position in the x axis. Using this fitness function, the individuals that moved forward will be rewarded, and the individuals that moved backward will be punished, receiving a negative fitness.

We started believing that with this fitness function, the individuals selected to produce offsprings would be the ones that have a more stable gait: a stable individual should to move longer than the one that fell down. But due to the fact that in the first generations almost all the individuals are unstable and fall down, the selected individuals were the ones that simply fell down in the forward direction. So, the selected individuals are not the individuals that can remain in the upright position and walk longer during the simulation. In this way, the Equation 2 didn't lead us to a good solution, and thus the GA makes an almost random search in the search space.

To avoid this problem, we developed an other fitness function that use sensorial information in order to make the gait learning more efficient. One of the less expensive and simpler robotic sensors are the bumpers. These contact sensors can be installed under the robot paws, and they indicate when the paw is touching the ground. Thus, we decided to simulate bumpers under the robot paws, and then it was possible to determine how many paws are maintained in contact with the ground for each instant of time. The new fitness calculation was accomplished through the equation:

$$F = D * \mu_P\,, \tag{3}$$

where μ_P is the average number of paws touching the ground. Using this fitness function, we noticed that an odd behavior began to happen. The individuals that maintained all the paws in the ground and just inclined forward the front of their bodies were rewarded more than those that lifted the paws from the ground during the walk. Thus, other modifications were accomplished in the

previous fitness function. This new fitness function is calculated through the equation:

$$F = D/(1 + B)\,, \tag{4}$$

where B is calculated through the equation:

$$B = (\mu_P - L/2)^2\,, \tag{5}$$

where L is the number of the robot legs. In this way, the individuals that maintain approximately half of the paws in contact with the ground, during walk simulation, will consider in the fitness computation the total distance covered by the robot. The individuals that fell down or didn't move the paws from the ground will be punished, receiving lower fitness values. This type of fitness function is more indicated for a *trot* gait.

Using the Equation 4, the gait learning became much more efficient, but we still had a lot of totally unstable solutions. The visualization of the obtained solutions showed unstable robots where the body height and inclination varies a lot during the walking simulation. Thus, besides the bumpers, we decided to add simulated inertial sensors (gyroscope). These sensors are used in some modern mobile robots and they are becoming a largely adopted device in walking machines. During the walk, readings from the simulated gyroscope are collected, and the robot instability measure G (Gyro) is calculated through the equation[10]:

$$G = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x}_x)^2 + \sum_{i=1}^{N}(y_i - \overline{x}_y)^2 + \sum_{i=1}^{N}(z_i - \overline{x}_z)^2}{N}}\,, \tag{6}$$

where N is the number of sample readings, x_i, y_i and z_i are the data collected by the simulated gyro at the time i, and $\overline{x}_x$, $\overline{x}_y$ and $\overline{x}_z$ are the mean values of gyro readings. The fitness function F is then calculated through the equation:

$$F = \frac{D}{1 + G + B}\,. \tag{7}$$

Analyzing the fitness function, we can observe that B reaches its smaller value when the robot maintains half of its endpoints (paws) touching the ground. This condition is desirable when the type of gait adopted is the *trot*. In this way, the best solutions have B close to zero. So, this parameter will have a strong influence in the population evaluation and evolution. Related to the other fitness parameters, the individual better qualified will be the one that has the best relationship between velocity and stability. The best solutions are those that moves fast, but without losing the stability[10]. After we included the instability measure G in the fitness function, we analyzed if it was possible to remove the average number of endpoints touching the ground from the fitness function. This new fitness function is represented in the following equation:

$$F = D/(1 + G)\,. \tag{8}$$

In Section 5 we describe the executed experiments using these four different fitness functions (Equations 2, 4, 7 and 8), and using the robots with four paws. These experiments were executed in order to verify which of these fitness functions is more efficient to generate stable gaits for legged robots.

4 Proposed System

The LegGen System[3][7] was developed to accomplish the gait control of simulated legged robots in an automatic way. It was implemented using the C++ programming language and the free software libraries ODE and GAlib. The LegGen System works as follows: initially the file describing the robot is loaded, and the robot is created in the ODE environment according to file specifications. After this, the system parameters are loaded, and the Genetic Algorithm is initialized and executed until the number of generations is reached. The evaluation of each chromosome is realized in the following way:

- The simulated robot is placed in the starting position and orientation;
- The genome is read and the robot control FSM table values are set;
- The physical simulation is executed during a predefined time (60 seconds in our experiments);
- Fitness is calculated and returned to the GAlib;

During the simulation, if all paws of the robot leave the ground at same time for more than one second, the simulation of this individual is immediately stopped, because this robot probably fell down, and therefore it is not necessary to continue the physical simulation until the predefined end time.

4.1 Modeled Robots

According to the documentation, computational complexity when using the ODE library is $O(n^2)$, where n is the amount of bodies present in the simulated physical world. Thus, in order to maintain the simulation speed in an acceptable rate, we should use few and simple objects. For this reason, all the simulated robots were modeled with simple objects, as rectangles and cylinders, and they have only the necessary articulations to perform the gait. Thus, body parts as the head and the tail are not present in the modeled robots. In order to keep our

	Dimensions		
Part	**x**	**y**	**z**
Body	45.0cm	15.0cm	25.0cm
Thigh	5.0cm	15.0cm	5.0cm
Shin	5.0cm	15.0cm	5.0cm
Paw	8.0cm	5.0cm	9.0cm

Fig. 1. Robot model used in the simulations

robot project simple, the joints used in the robots legs just move around the z axis of the robot (the same axis of our knees), and the simulations just used robots walking in a straight line. In the near future, we plan to extend our system to accept more complex robot models and joints. Several robot types were

[3] LegGen – http://www.inf.unisinos.br/~osorio/leggen

developed and tested, before we defined the final main model, showed in the Figure 1, with four legs and three parts per leg. The simulated robots dimensions are approximately the dimensions of a dog.

5 Results

In order to determine the best fitness function, several experiments were conducted using the four fitness functions described in Section 3.5 (Equations 2, 4, 8, and 7). Our main objective was to discover which fitness function represents the best relation between stability (better gait performance) and cost (less hardware expenses).

Our intention is to create a real robot using the fitness function that better improves the gait control learning by the Genetic Algorithm, but with an accessible final cost. The Equations 8 and 7 need a gyroscope sensor to compute the fitness function, so these solutions are more expensive. The Equation 4 needs just bumper sensors, which have a quite accessible cost and can be easily used in a real robot. Thus, the use of gyroscope sensors would be justifiable only if they present a significant increase in the robot performance in terms of speed and stability. In relation to the total number of robot legs, the best choice is to use the smallest possible number of legs that allows a stable gait.

The Table 1 describes the results obtained in the four different type executed experiments. Due to stochastic nature of the Genetic Algorithms, each experiment described in Table 1 was repeated 30 times using different random seeds, and the mean and standard deviation values relative to the fitness function and sensors information obtained from these experiments were calculated.

Table 1. Results obtained in the simulations

		F		D		G		B	
Exp	Fitness	μ	σ	μ	σ	μ	σ	μ	σ
E01	Equation 2	205.81	76.07	432.55cm	114.56cm	1.42	1.26	0.12	0.60
E02	Equation 4	225.29	51.60	439.63cm	57.32cm	1.02	0.42	0.01	0.02
E03	Equation 8	261.72	42.04	440.60cm	64.48cm	0.69	0.13	0.01	0.01
E04	Equation 7	268.77	37.84	454.55cm	68.48cm	0.69	0.17	0.01	0.01

The first column indicates the experiment identification, the second indicates the fitness function used, the third and fourth columns show, respectively, the mean (μ) and the standard deviation (σ) of the fitness function (F). The fifth and sixth columns show the μ and the σ of the distance covered by the robot (D) in centimeters, the seventh and eighth columns show the μ and the σ of the robot instability (G), and the last two columns show the μ and the σ of the average number of endpoints in the ground (B). The Figure 2 shows the boxplot graph and the 90% confidence interval of the Table 1 results.

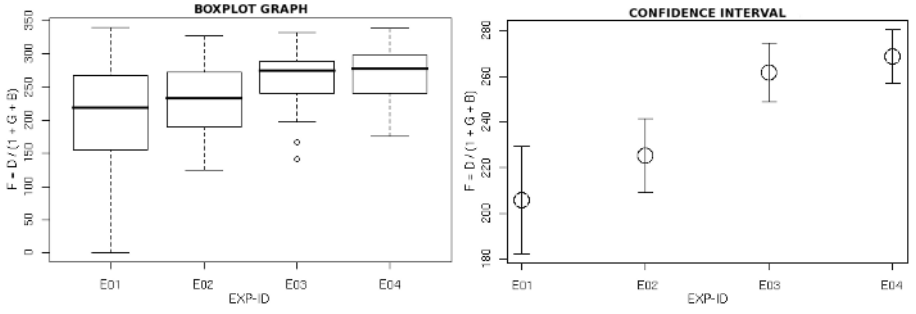


Fig. 2. Statistical analysis of results

From the observed results presented in Table 1, we can reach the conclusion that the distance covered by the robots in the experiment E04 are greater than the distance covered by the robots in the others experiments.

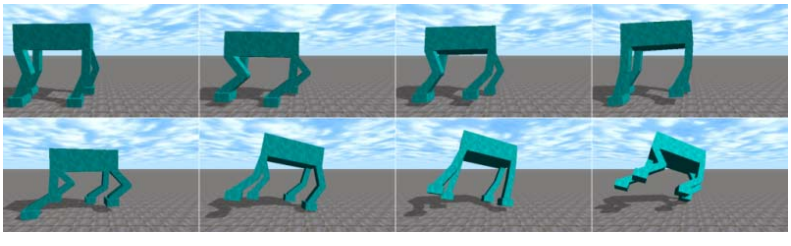


Fig. 3. Example of a generated gait (experiment 01)

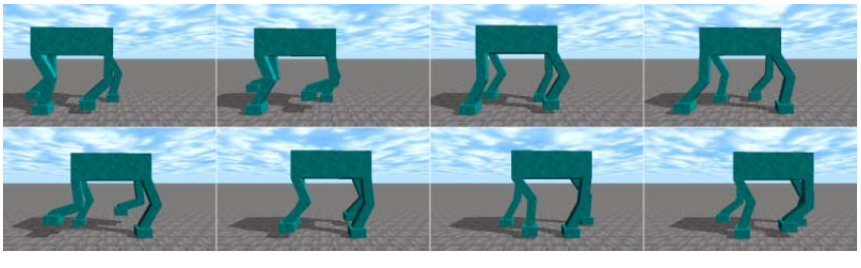


Fig. 4. Example of a generated gait (experiment 04)

Considering the instability, the use of gyroscope sensor (E03) increased the stability more than the use of the bumper sensors (E02). The Figure 3 shows an example of generated gait obtained in the experiment E01, and the Figure 4 shows an example of generated gait obtained in the experiment E04[4].

[4] Some videos are available in `http://www.inf.unisinos.br/~osorio/leggen`

6 Conclusions and Perspectives

Based on the performed experiments, we observed that fitness functions with additional sensorial information are very useful to generate stable gaits. We also concluded that although these gaits can be slower than gaits generated with simple fitness functions (few sensorial information), we are able to obtain more stable gaits.

The perspectives of this work includes to adapt gait control in order to make possible control robots moving over irregular surfaces and to climb or to descend stairs, as well as, this work will help us in the physical robot construction based on the specifications of our best learned models. The real robot implementation created from a virtual model will help us to validate the control system in real conditions. We are also using the implemented system to test other robot configurations with different number of legs (4, 6, 8) and joints orientation (forward, backward) with promising results.

References

1. Dudek, G., Jenkin, M.: Computational Principles of Mobile Robotics. Cambridge Univ. Press, Cambridge, UK (2000)
2. Wyeth, G., Kee, D., Yik, T.F.: Evolving a locus based gait for a humanoid robot. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS). Volume 2., Las Vegas, NV (2003) 1638–1643
3. Chernova, S., Veloso, M.: An evolutionary approach to gait learning for four-legged robots. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), Sendai, Japan (2004)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)
5. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA (1996)
6. Darwin, C.: Origin of Species. John Murray, London, UK (1859)
7. Heinen, M.R., Osório, F.S.: Applying genetic algorithms to control gait of physically based simulated robots. In: Proc. IEEE Congr. Evolutionary Computation (CEC), Vancouver, Canada (2006)
8. Holland, J.H.: Adaptation in Natural and Artificial Systems. Univ. Michigan Press, Ann Arbor, MI (1975)
9. Wolff, K., Nordin, P.: Evolutionary learning from first principles of biped walking on a simulated humanoid robot. In: Proc. Advanced Simulation Technologies Conf. (ASTC), Orlando, FL (2003)
10. Golubovic, D., Hu, H.: Ga-based gait generation of sony quadruped robots. In: Proc. 3th IASTED Int. Conf. Artificial Intelligence and Applications (AIA), Benalmadena, Spain (2003)

Does Complex Learning Require Complex Connectivity?

Carlos Rubén de la Mora-Basáñez[1,2],
Alejandro Guerra-Hernández[1], and Luc Steels[2]

[1] Departamento de Inteligencia Artificial
Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Sebastián Camacho No. 5, Xalapa, Ver., México, 91000
{cdelamora, aguerra}@uv.mx
[2] Artificial Intelligence Laboratory
Vrije Universiteit Brussel
Pleinlaan 2, 10G725, Brussels, Belgium 1070
luc@arti.vub.ac.be

Abstract. Small World and Scale Free network properties characterize many real complex phenomena. We assume that low level connectivity with such topological properties, e.g., anatomical or functional connectivity in brains, is compulsory to achieve high level cognitive functionality, as language. The study of these network properties provides tools to approach different issues in behavior based Artificial Intelligence (AI) that usually have been ill defined, e.g., complexity and autonomy. In this paper, we propose a model in which situated agents evolve knowledge networks holding both Small World and Scale Free properties. Experimental results in the context of Pragmatic Games, elucidate some required conditions to obtain the expected network properties when performing complex learning.

1 Introduction

Behavior based AI [6,21,23] claims the necessity to work with autonomous, embodied and situated systems, always arguing biological referents. Similar approaches include development [28,19,15] as an ineluctable issue. However, concepts such as embodiment [27], autonomy [7], or even the epigenetic [13,14], are ill defined or even are used with different connotations.

The validation of behavior based AI systems involves comparisons with the functionality of observed systems, being difficult to define general measures. Despite that, a kind of universal way of organization in connectivity has been recently found in a huge number of real complex phenomena: the complex networks [26,4,2,18] exhibiting Small-World and Scale-Free properties.

Complex networks are present at different levels of cognitive functionality, from functional connectivity in brains [10], to high level cognitive behavior as language [11]. Although not yet well understood, this means that high level functionality is grounded on appropriate low level connectivity.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 572–581, 2006.

In this paper, we present a model in which an artificial situated agent develops knowledge networks exhibiting complex topologies. Experimental results, using Pragmatic Games [8], support explanations about the required conditions to obtain networks with the expected properties, i.e., Small World and Scale Free, relating degree distribution and sensing; clustering coefficient and biological motivations; goals, acquired knowledge, and attentional focus. This constitutes a relevant advance in the understanding of how may low level connectivity emerge in artificial agents.

Organization is as follows: Section 2 introduces complex networks. Section 3 presents the methodology of Pragmatic Games. Section 4 discusses experimental results. Finally, section 5 concludes the paper.

2 Complex Networks

Many interesting phenomena involving networks with a huge number of nodes does not hold a random topology, and their degree distribution does not follow a normal distribution. Such networks hold a complex topology [26,4,2,18], where characteristic parameters are statistical macro variables describing the whole network and representing a fingerprint of complexity. Small World networks exhibit short characteristic path lengths but high clustering coefficients, when compared to random networks. These measures are used to determine how far of randomness, or near to order, a network is. Interesting networks were found in the middle between randomness and complete order. Scale Free networks exhibit power-law degree distributions, rooted in their growth mechanism [3].

Phenomena holding the Small World and Scale Free properties include social, informational, technological, and biological networks, in fields as diverse as the WWW [1], metabolic networks [12], human language [11], scientific collaboration networks [5], brain networks in mammalians [25], or functional brain networks in humans [10]. Clear examples of how high level cognitive functions emerge from low level connectivity are present in brain connectivity, particularly in human cerebral cortex [10], and also in other mammalians [25]; and in language networks [22].

The Scale Free property of brains explain the coexistence of functional segregation and integration, as well as redundancy and efficiency in information transmission. The power-law in degree distribution affects the functional impact of brain lesions: few highly connected nodes are vulnerable to damage, which explains de small impact of damage on random lesions. Cortical areas in mammalian brains exhibit attributes of complex networks. The distribution of functional connections and the probability of finding a link versus distance are both Scale Free. Additionally, the characteristic path length is small and the clustering coefficient is high, when compared with random graphs. Short path length captures potential functional proximity between regions. High clustering measures the degree of which a particular area is part of local collective dynamics. Frequent connectivity in all shortest paths linking areas, explains structural stability and efficient working of cortical networks.

Human language is an important example of complex cognitive functionality with complex networks properties, mounted on complex connectivity (the brain itself). This connectivity allows the fast and robust construction of a huge variety of sentences from limited number of discrete units (words). Co-ocurrence, syntactic, and semantic networks shown to hold complex connectivity [22], having power-law degree distributions ($\gamma \sim 2.2 - 3.0$), high clustering coefficient ($C/C_{rand} \sim 10^3$), and small path-length. Hub deletion have the effect of loosing optimal navigation in co-occurrence networks; articulation loss in the case of syntactic networks; and conceptual loose plasticity in the case of semantic networks. Graphs of word interactions [11] show small-world properties (high clustering coefficient) and scale-free distributions ($\gamma \sim 1.5 - 2.0$). Based on the significance profile of small sub-graphs from different languages, common local structure in word adjacency networks is also observed [17]. This allows the definition of universal classes of networks by local motif statistics [16].

The study of complex networks is an active field which aims to understand complex phenomena, focusing in finding statistical properties to characterize the structure and behavior of such networks; creating models to explain such parameters; and predicting their behavior based on topological properties and local rules for nodes and links. In particular, we are interested in the necessary conditions that enables a situated agent to develop complex networks and how to obtain them.

3 Pragmatic Games

Pragmatic Games [8], inspired in language games [24], are used to explore how an artifical agent can develop complex knowldge networks. They offer a methodology by which situated agents are immersed in similar but not identical repetitive situations to perform complex spatial learning. With these games we are able to control, test, and make measurements of the network development. Three games (Fig. 1) are defined:

- Focusing game. One agent having only one eye and its visual field. The game starts setting an object within an environment in a random place. The eye moves randomly, if the fovea "sees" the object, the game is restarted.
- Grasping game. One agent having one eye and one hand. The game restarts when the hand reaches the object.
- Feeding game. One agent having one eye, one hand, and a mouth. When the hand passes over the object, the agent closes its hand and the object is attached to it. The random movements continue until the hand (with the object) reaches the mouth. At this moment, the game restarts.

The knowledge acquired by the agents while performing complex learning is represented as an evolving network of affective states, processed by a mechanism emulating epigenesis (see subsection 3.2). The morphology of the agent, the specific interaction within the environment (the game), together with the mechanism proposed to grow up the networks, establish the minimal conditions discriminating the games that produce complex networks and those that do not.

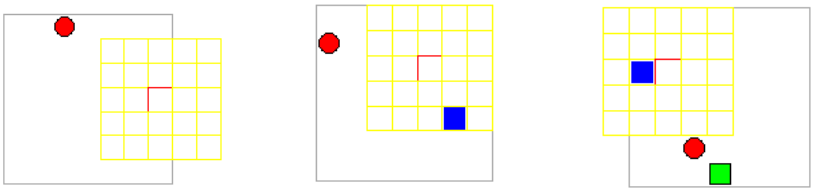


Fig. 1. Pragmatic Games, from left to right: Focusing (visual field), Grasping (visual field and hand) and Feeding (visual field, hand, and mouth). In all games the object is the circle.

3.1 The Agent

Our agent has 77 bit sensors: 75 bits for a 5×5 red (R), green (G), and blue (B) sensitive visual field; one bit for a 1×1 blue "hand"; and one for a 1×1 green "mouth". The agent has four actuators: two for the eye and two for the hand. Each actuator has three possible states: do nothing, up/right, and down/left. The mouth is always fixed. An actuation is specified by a set of four values $(e_x, e_y, h_x, h_y \in \{1, 0, -1\})$. The agent moves randomly choosing one of the three possible states of each one of the actuators, considering these states as random variables with uniform distribution.

The agent has a set of distinguishable innate affective states [20], called biological motivations. These biological motivations are mapped to a 5-bit vector: Three bits for detecting RGB in the fovea; one for the hand holding an object, and other for the presence of the object in the mouth. Therefore, there are 32 possible biological motivations. These biological motivations do not have any appetitive or aversive character, they are only distinguishable. At the beginning they are not related with any sensorial state, the relation must be established by the mechanism which makes the network grow up.

3.2 Network Growth Mechanism

Every time the agent experiences a particular biological motivation, this motivation and the sensing state (77 bit string) is saved. Thus, every biological motivation has associated a record of sensing vectors. This process does not affect the network. Any biological motivation may give place to an affective state or a potential affective state, which is incorporated to the network by two mechanisms: 1) Detecting affective states from biological motivations. The affective state represents the set of sensing bits always present in the associated record, every time the biological motivation has been experienced, or; 2) Detecting potential affective states. If the sensing state at time t corresponds to an affective state, then a node corresponding to the sensing state in the time $t-1$ is incorporated, as well as the directed arc between the nodes (representing the actuation).

A potential affective state could become affective state if its frequency exceeds some value. If the values are too small, noise can be learned. If the values are too

big, then it takes more time to learn. This also happens for other parameters of the model.

The arcs joining nodes are incorporated in the network in two ways: 1) When a potential affective state is reached, as mentioned above, and; 2) When the agent experiences two consecutive affective or potential affective states.

Arcs are labeled as frequent or codified. Once an arc exists between two nodes, some statistics (frequency and actuation) are updated if it is traversed. Its label is then computed in the following way:

1. If the frequency of occurrence for an arc is higher than a given value, it is labeled as a frequent arc. The distribution of probabilities for the actuators is computed from the history and saved in the arc in the form: $\{\ p(e_x = -1),\ p(e_x = 0),\ p(e_x = 1)\ \}$, $\{\ p(e_y = -1),\ p(e_y = 0),\ p(e_y = 1)\ \}$, $\{\ p(h_x = -1),\ p(h_x = 0),\ p(h_x = 1)\ \}$, $\{\ p(h_y = -1),\ p(h_y = 0),\ p(h_y = 1)\ \}$. The distributions are normalized.
2. If one of the three probabilities in each one of the four triplets of a frequent arc is higher than a threshold, all the triplet is replaced with one code, representing the associated winning movement for the correspondent actuator. For example, if the distribution of probabilities for the eye in the x direction is $\{\ p(e_x = -1),\ p(e_x = 0),\ p(e_x = 1)\ \} = \{\ 0.1,\ 0.3,\ 0.7\ \}$, then the triplet is replaced by a 1 ($p(e_x = 1) = 0.7$ is the highest probability over the threshold).

3.3 Attentional Focus

Attentional focus is a behavior modifier [9] given as a number between $[0, 1]$ representing the probability to undo the last movement performed. Thus, a 0.5 attentional focus means that the agent has a 0.5 of probability to revisit the last sensorial state. Each one of the pragmatic games was performed using different attentional focus values: 0.0, 0.25, 0.50, and 0.75. A 0.0 value is equivalent to random movement. A 1.0 value was not used, because it implies an infinite loop between two sensing states. These attentional focus values are fixed during all the experiments.

4 Results

Experiments showed that the networks evolved by the situated agents depend on the played game and the mechanism to make the network grow up. They also showed that wiring is made in terms of experiences. First, the complexity of interactions was found relevant to evolve complex networks, relating degree distribution and sensing. Second, biological motivations affect the clustering coefficient of the obtained networks. And third, attentional focus affects the number of goals achieved and the amount of knowledge acquired.

4.1 Degree Distribution and Sensing

Power-law degree distribution is a fingerprint to distinguish complex networks. This property reflects the existence of few highly connected nodes (hubs), and a lot of them few connected. This allows tolerance to random fails, but vulnerability if the failure affects a hub.

A substantial difference exists between unimodal and multimodal games. In the unimodal game (focusing) the resulting network is by no means complex. But in multimodal games (grasping and feeding) a power-law emerges (Fig. 2). The only difference between grasping and focusing games, is the 1×1 moving hand around the world, absent in the focusing game. It is not only the complexity of the environment what matters, but the complexity of the interactions.

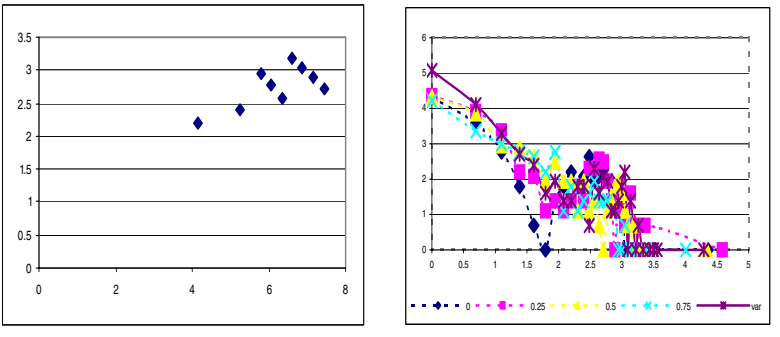


Fig. 2. Degree distribution for focusing game at left ($degree \times frec$) and grasping game at right ($\ln degree \times \ln frec$). The unimodal–multimodal character of the game affects the complexity of the resulting network.

4.2 Clustering Coefficient and Biological Motivations

Clustering coefficient gives a measure of the possibility of being part of a local collective dynamics. In some way, it measures the integration of elements in a network. It is shown that clustering coefficient in the multimodal games is high when compared with a random graph (Fig. 3). It is also shown that this measure depends on the growing mechanism itself, particularly on the consideration of biological motivations (Fig. 4).

4.3 Goals, Acquired Knowledge, and Attentional Focus

The number of played games in each run can be understood as the number of goals achieved by the agents. The experiments show that the rate of growth of the number of goals in time is similar for all the games. A lower focus, as expected, is associated with a higher probability to advance, resulting in more games played (Fig. 5).

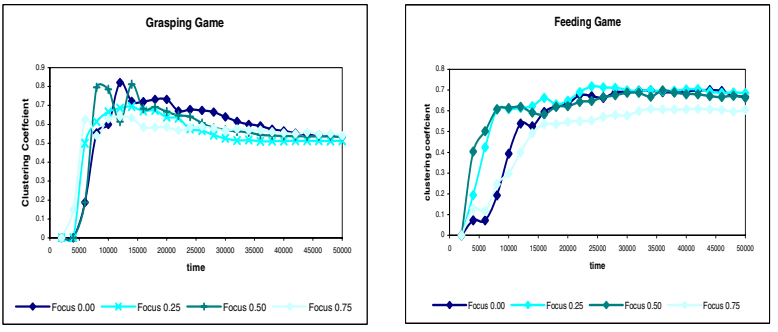


Fig. 3. Clustering coefficient in grasping game (left) and feeding game (right) is high when compared with an equivalent random graph

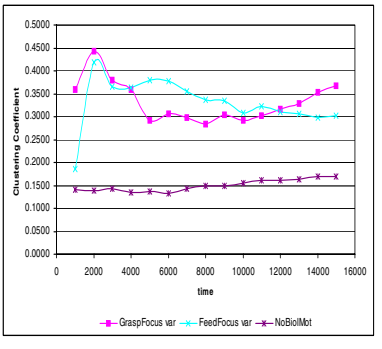


Fig. 4. Clustering coefficient and biological motivations. When biological motivations are not considered in the epigenetic mechanism to evolve the net, the clustering coefficient is very low.

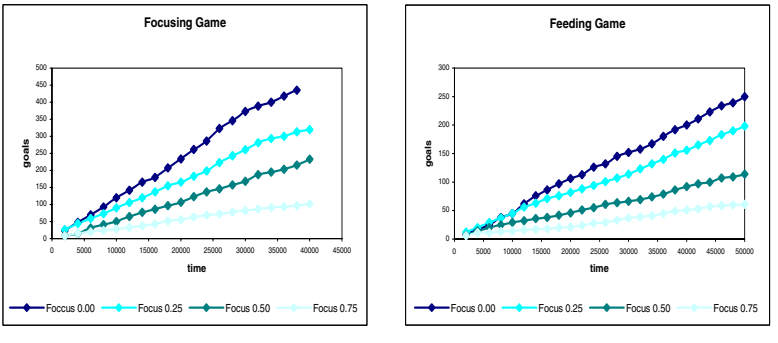


Fig. 5. The number of games played (goals) by the agent is similar when focus is changed, both for focusing and feeding games

However, this tendency is different when the amount of knowledge acquired is considered. A fact is identified as two affective states connected by a coded arc. The number of facts is a measure of the quantity of acquired knowledge. It depends on the focus value, but its effect varies with the played game.

In our unimodal game, more goals and knowledge are attained by the agent for lower focus values. In the multimodal games the opposite is observed. This seems to be the source of the well known exploration/exploitation trade–off in learning agents. If the game is multimodal, the agent requires different behavior to achieve goals and to acquire knowledge. In the unimodal game, this situation does not exist (Fig. 6).

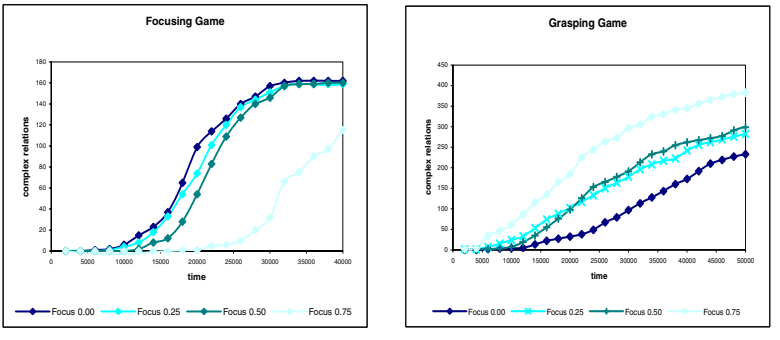


Fig. 6. The knowledge acquired depends both on the game played and the focus value

5 Conclusions

We use pragmatic games as a methodology to study knowledge development in situated agents. Knowledge is represented as an evolving network resulting from a mechanism emulating epigenesis, which is enacted by experiences. The knowledge acquired in this way is complex, because it relates spatial rules in appropriate way. The proposed mechanism to evolve knowledge includes innate biological motivations recognition, and the incorporation of nodes and links in steps. Network analysis and the associated measures are applied on the corpus of acquired knowledge.

In our experiments, clustering coefficient is large (a Small World property), compared with random graphs. This result is due to the proposed epigenetic mechanism. Particularly, because the innate biological motivations starting the process give rise to hubs in the network.

Knowledge networks in games involving multimodal agents exhibit power-law distribution (Scale Free property), whereas the games played by unimodal agents do not. This means that complex learning arises independently of the complexity of the environment, but depending on the interaction the agent has with it. Connectivity distribution in our unimodal game is finite and well defined,

no matter the size of the eye. If the hand is introduced, the number of states increases significantly, emerging a complex network.

To observe how might the agent behavior affect acquired knowledge, we use a behavior modulator called attentional focus. If we consider the number of achieved games as the number of goals achieved by the agent, we observe that the focus value have the same impact for both unimodal and multimodal games, i.e., lower focus values result in more achieved games. However, the number of incorporated facts in the network depends on the modality of the game. For the unimodal game, lower focus values result in higher number of facts; but for the multimodal games, the inverse relation is observed. Thus, the modality of the game seems to be the source of the well known exploitation-exploration trade-off in learning agents.

The epigenetic mechanism used to make grow the network is suitable, in the sense that the topology of the generated networks evolves showing complex network properties. These results suggest that artificial agents must consider biological motivations in their learning mechanisms and be multimodal.

Complex connectivity in artificial agents seems to be necessary, because properties considered as essential in the definition of such agents, emerge in natural way from complex connectivity. For an artificial agent, complex topology represents the potentiality to exhibit complex functionality, i.e, the dynamics that becomes exploitation of the acquired knowledge.

Future work will consider cognitive autonomy which enables the agent to adapt its behavior to maximize knowledge acquisition, so that the amount of acquired knowledge becomes a behavior modulator. The effect of the relations between knowledge and behavior will be quantified and analyzed in terms of complex networks properties.

References

1. R. Albert, H. Jeong and A-L. Barabási, The diameter of the World Wide web, *Nature*, 401:130–131 (1999)
2. R. Albert and A-L. Barabási, Statistical Mechanics of Complex Networks, *Reviews of Modern Physics*, 74, 47 (2002)
3. A-L. Barabási and R. Albert, Emergence of scaling in random networks, *Science*, 286:509–512 (1999)
4. A-L. Barabási, **Linked: The New Science of Networks**, Perseus (2002)
5. A-L. Barabási et al., On the topology of the scientific collaboration networks, *Physica A*, 311:590–614 (2002)
6. R.A. Brooks, **Cambrian Intelligence: The early history of the New AI**, MIT Press (1999)
7. J. Collier, Autonomy and Process Closure as the basis for functionality, *Annals of the New York Academy of Sciences*, J.L.R. Chandler and G. Van der Vijver (eds.), 901:280–290, New York, USA (2000)
8. C. R. de la Mora-Basáñez, C. Gesherson, and V. A. García-Vega. Representation development and behavior modifiers. In C. Lemaître, C. A. Reyes, and J. A. González, editors, *Adavances in Artificial Intelligence: Iberamia 2004*, volume 3315 of *LNAI*, pages 504–513, Berlin, Germany, 2004. Iberamia, Springer Verlag.

9. D. Dorner and K. Hille. Artificial souls: Motivated emotional robots. In *IEEE Conference Proceedings, International Conference on Systems, Man, and Cybernetics; Intelligent Systems for the 21st Century* volume 4, *pages 3828–3832, Vancouver, Canada, 1995*
10. V.M. Eguiluz et al., Scale-free brain functional networks, *Phys. Rev. Letters*, 92, 018102 (2005)
11. R. Ferrer-i-Cancho and R.V. Sole, The small world of human language, *Proceedings of The Royal Society of London*, Series B, Biological Sciences, 268(1482):2261–2265, UK (2001)
12. H. Jeong et al., The large-scale organization of metabolic networks, *Nature*, 407:651–654 (2000)
13. J. Lederberg, The Meaning of Epigenetics, *The Scientist* 15(18):6 (2001)
14. J.M. Levenson and D. Sweatt, Epigenetic mechanisms in memory formation, *Nature* 6:109–119, February (2005)
15. A. Mcintyre, F. Kaplan and L. Steels, Crucial Factors in the Origins of Word-Meaning, *The Transition to Language*, A. Wray (ed.), pages 252–271, Oxford University Press, Oxford, UK (2002)
16. R. Milo et al., Network Motifs: Simple Building Blocks of Complex Networks, *Science* 298:824–827, October (2002)
17. R. Milo et al., Superfamilies of evolved and Designed Networks, *Science* 303: 1538–1542, Mach (2004)
18. M.E.J. Newman, The structure and function of complex networks, *SIAM Review*, 45(2):167– 256 (2003)
19. C.G. Prince, Introduction: The Second International Workshop on Epigenetic Robotics, Lund University Cognitive Studies 94, Lund, Sweden (2002)
20. M. Scheutz, The Evolution of Simple Affective States in Multi-Agent Environments, *Proceedings of AAAI Fall Symposium 01*, AAAI Press, USA (2001)
21. T. Smithers, Taking Eliminative Materialism Seriously: A Methodology for Autonomous Systems Research, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, F.J. Varela and P. Bourgine (eds.), MIT Press / Bradford Book, pages 31–40 (1992)
22. R.V. Sole et al., Language Networks: their structure, function and evolution, Santa Fe Institute Working Paper 05-12-042, Santa Fe, CA., USA (2006)
23. L. Steels and R. Brooks, **The Artificial Life route to Artificial Intelligence: Building situated embodied agents**, Lawrence Erlbaum Ass. (1995)
24. L. Steels, Self-organizing vocabularies, *Proceedings of Alife V*, C. Langton and T. Shimohara (eds), pages 179–184, Nara Japan (1996)
25. O. Sporns et al.,Organization, development and function of complex brain networks, *Trends in Cognitive Sciences*, 8:418–425 (2004)
26. D. Watts, **Small-Worlds. The dynamics of networks between order and randomness**, Princeton Studies in Complexity, Princeton University Press, USA (1999)
27. T. Ziemke, Are Robots Embodied?, *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Lund University Cognitive Studies 85, Lund, Sweden (2001)
28. J. Zlatev and C. Balkenius, Introduction: Why 'epigenteic robotics'?, *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Lund University Cognitive Studies, 85:1–4, Lund, Sweden (2001)

The Predicate-Minimizing Logic MIN*

Francicleber Martins Ferreira and Ana Teresa Martins

Departamento de Computação,
Universidade Federal do Ceará,
CP 12.166, Fortaleza-CE, Brasil, 60455-760
{fran, ana}@lia.ufc.br

Abstract. The concept of minimization is widely used in several areas of Computer Science. Although this notion is not properly formalized in first-order logic, it is so with the logic MIN(FO) [13] where a minimal predicate P is defined as satisfying a given first-order description $\phi(P)$. We propose the MIN logic as a generalization of MIN(FO) since the extent of a minimal predicate P is not necessarily unique in MIN as it is in MIN(FO). We will explore two different possibilities of extending MIN(FO) by creating a new predicate defined as the union, the U-MIN logic, or intersection, the I-MIN logic, of the extent of all minimal P that satisfies $\phi(P)$. We will show that U-MIN and I-MIN are interdefinable. Thereafter, U-MIN will be just MIN. Finally, we will prove that simultaneous minimizations does not increase the expressiveness of MIN, and that MIN and second-order logic are equivalent in expressive power.

1 Introduction

The concept of minimal objects is widely used in several areas of Computer Science as, for instance, Programming Languages, Theory of Computation and Artificial Intelligence. We say that an object in a set is *minimal* with respect to a certain order, if there is no other object in this set less than it. If there is only one minimal object and it is less than or equal to any other object, it is called the *least* element, the *minimum* [5].

In Denotational Semantics of Programming Languages, we can define recursive functions as the least element of a domain D, *i.e.*, the least fixed point of a chain induced by a certain functional F over D [12]. As an illustrative example [11], the factorial function, fac, over the set of natural numbers, Nat, is the least fixed point of the functional $F : (Nat \to Nat_\perp) \to (Nat \to Nat_\perp)$ defined in lambda notation as $F = \lambda f.\lambda n.\ n\ equals\ zero \to one\ []\ n\ times\ f(n\ minus\ one)$.

One of the models of computability is the class of recursive functions [1]. We say that a function is computable if and only if it can be obtained from the initial functions by a finite number of applications of composition, recursion and proper minimalization. The minimalization of a predicate $P(x_1, \ldots, x_n, y)$, $min_y P(x_1, \ldots, x_n, y)$, is the least value of y for which the predicate

* This research is partially supported by PADCT/CNPq, PRONEX/FUNCAP-CNPq, CNPq and FUNCAP.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 582–591, 2006.

P is true, if there is one. Otherwise, $min_y P(x_1, \ldots, x_n, y)$ is undefined. When $min_y P(x_1, \ldots, x_n, y)$ is a total function, that is, when for each $x_1, \ldots, x_n$ there is at least one y for which $P(x_1, \ldots, x_n, y)$ is true, we say that we are applying proper minimalization to P. Hence, the characterization of a computable function strongly depends on this notion of minimalization.

The idea of minimal model is also essential for logic programming. A minimal Herbrand model for a logic program is the one where all predicates defined by program clauses have their minimal extension. Such a model is assured to exist due to the universal Horn-clause syntax of logic programs [8].

A more general notion of minimal predicate appear in the theory of Circumscription [9,10] used for formalizing aspects of nonmonotonic reasoning in Artificial Intelligence. In case of predicate circumscription, the idea is to find out the minimal extent of a predicate P that appear in a set of axioms Φ, possibly varying the interpretation of some designated object, function, and predicate constants that appear in Φ, and keeping the interpretation of all other non-designated object, function, and predicate constants fixed. The concept of Closed World Assumption in Deductive Databases is precisely expressed by using predicate circumscription.

It is clear that first-order logic cannot express the notion of predicate minimization. However, is there any deductive, monotonic logic able to formalize the concept of predicate minimization properly? The answer is affirmative, and we can give two examples: the logics LFP and MIN(FO). The logic LFP (Least Fixed Point) [6] extends the first-order logic with predicates that are equivalent to the least fixed point of some operators. Such fixed points always exist due to the fact that these operators are monotone. Within LFP, we can define many tractable graph properties such as graph connectivity, reachability, and also the graph of recursive arithmetic functions on successor structures.

An alternative approach to define minimal predicates is introduced by the logic MIN(FO) [13]. Instead of using fixed point on monotone operators, the unique existence of a minimal predicate P satisfying a given first-order description $\phi(P)$ is assured by the fact that $\phi(P)$ must follow a special first-order syntactic pattern, the PIA conditions. Thus, iterated predicate minimization on PIA-conditions results in a language MIN(FO) that is equivalent in expressive power to LFP. The predicate-minimizing logic MIN(FO) will be presented in section 2.

The logic MIN is here proposed as a generalization of MIN(FO). While the definition of a minimal predicate P results in the minimum one in MIN(FO), the extent of a minimal predicate P is not necessarily unique in MIN. As in MIN(FO), we can iterate predicate minimization in MIN, but the formula $\phi(P)$ need not to follow any specific syntactic pattern as in MIN(FO). In the third section, we will introduce the U-MIN and I-MIN logics and prove their expressive equivalence. Thereafter, U-MIN will be called just MIN. In section 4, we will investigate the notion of simultaneous minimizing and MIN will be characterized as equivalent to second-order logic. Finally, in the last section, we will resume what we have done and point out some future work. In what follows, S is a

symbol set (that is, a set of relation, function and constant symbols), an S-structure $\mathfrak{A}$ is a pair consisting of a universe A and a function which associates to each n-ary relation symbol an n-ary relation on A, to each n-ary function symbol an n-ary function on A and to each constant symbol an element of A. An S-interpretation $\mathfrak{I}$ is a pair consisting of an S-structure $\mathfrak{A}$ and an assignment β which associates an element of A to each variable. The notation and first-order logic basic results used in this text are detailed in [3].

2 The Predicate-Minimizing Logic MIN(FO)

The predicate-minimizing logic MIN(FO) was introduced in [13] as an alternative approach to LFP. We will first present MIN(FO) without iterated predicate minimization, give a completeness result of van Benthem w.r.t. the syntactic PIA-format, and introduce the whole language of MIN(FO) with nested applications of predicate minimization. All theorems are proved in [13]. In what follows, let P be a predicate letter, $\overline{Q}$ a tuple of predicate letters, $\overline{t}$ a tuple of terms, and $\overline{x}$ a tuple of individual variables.

A minimal predicate $\mathbf{P}$ is uniquely defined in a model $\mathfrak{A}$ as the smallest $\mathbf{P}$ satisfying a certain first-order description $\phi(P, \overline{Q})$. The general scheme is:

$$[MINP \bullet \phi(P, \overline{Q})](\overline{t}), \text{ where } \phi(P, \overline{Q}) \text{ is a first-order formula.}$$

The unique existence of minimal predicates is explained by a model-theoretic criterion: the Intersection Property.

Definition 1 (Intersection Property). *A first-order formula $\phi(P, \overline{Q})$ has the intersection property for P if, in any structure $\mathfrak{A}$, whenever $(\mathfrak{A}, \mathbf{P}_i) \models \phi(P, \overline{Q})$ for all predicates in a family $\{\mathbf{P}_i | i \in I\}$, ϕ also holds for their intersection $(\mathfrak{A}, \bigcap_{i \in I} \mathbf{P}_i) \models \phi(P, \overline{Q})$.*

The syntactic counterpart of the Intersection Property is the PIA condition, a sort of generalized Horn clause.

Definition 2 (PIA Condition). *A first-order formula with identity is a PIA condition if it has the syntactic form $\forall x(\psi(P, \overline{Q}, \overline{x}) \rightarrow P(x))$, with P occurring only positively in the antecedent formula $\psi(P, \overline{Q}, \overline{x})$.*

The main result states that the syntactic PIA-format is expressively complete to assure Intersection Property, that is:

Theorem 1 (van Benthem). *The following are equivalent for all first-order formulas $\phi(P, \overline{Q})$:*

1. *$\phi(P, \overline{Q})$ has the Intersection Property w.r.t. P;*
2. *$\phi(P, \overline{Q})$ is definable by means of a PIA formula w.r.t. P.*

The MIN(FO) language allows nested applications of predicate minimization by the use of extended PIA conditions, namely:

Definition 3 (Language of MIN(FO)). *The language of MIN(FO) and the set of* extended PIA-conditions *are defined by simultaneous induction as the least sets such that: (i) every first-order formula is a MIN(FO) formula; (ii)every PIA-condition is an extended PIA-condition; (iii)whenever there is an extended PIA-condition $\phi(P, \overline{Q})$ and a tuple of terms $\overline{t}$ with the same lenth as the arity of P then $[MINP \bullet \phi(P, \overline{Q})](\overline{t})$ is a MIN(FO)-formula; (iv) every MIN(FO)-formula of the form $\forall x(\psi(P, \overline{Q}, \overline{x}) \rightarrow P(\overline{x}))$ such that P occurs only positively in $\psi(P, \overline{Q}, \overline{x})$ is an extended PIA-condition. (A relational symbol R occurs only positively in a MIN(FO) formula $[MINP \bullet \phi](t)$, for $R \neq P$, iff it occurs only positively in ϕ.)*

MIN(FO) was proved to be an alternative approach to least fixed point logics such as LFP [6]. The language of LFP extends the usual formation rules for first-order syntax with an operator defining smallest fixed points:

$$[\mathbf{lfp}P, \overline{x} \bullet \phi(P, \overline{Q}, \overline{x})](\overline{t}),$$

where P may occur only positively in $\phi(P, \overline{Q}, \overline{x})$, $\overline{x}$ and $\overline{t}$ are tuples with the same length as the arity of P.

The semantics of this new formula is: $\mathfrak{A} \models [\mathbf{lfp}P, \overline{x} \bullet \phi(P, \overline{Q}, \overline{x})](\overline{a})$ iff $\overline{a}$ is in the least fixed point of F_ϕ, where $F_\phi : \wp(A^k) \rightarrow \wp(A^k)$ is the monotone operator induced by $\phi(P, \overline{Q}, \overline{x})$ defined by $F_\phi(X) = \{\overline{a} \mid \mathfrak{A} \models \phi(P, \overline{Q}, \overline{x})[X, \overline{a}]\}$. Since, by the Tarski-Knaster theorem [6], every monotone operator has a fixed point, the semantics of this new formula is well defined.

Finally, van Benthem also established the following result [13]:

Theorem 2 (van Benthem). *MIN(FO) and LFP have equal expressive power, that is, for each MIN(FO) formula there is a LFP formula with the same models and vice-versa.*

3 Extending MIN(FO): The MIN Logic

In this section, we will define the MIN logic as a generalization of MIN(FO). From now on, let S be a symbol set, $\mathfrak{A}$ an S-structure used to interpret all symbols in S, $\mathfrak{I} = (\mathfrak{A}, \beta)$ an S-interpretation (or, simply, interpretation) where β is an assignment of the variables to elements in $\mathfrak{A}$. The definition of the satisfaction relation is as in [3]. Let us now introduce some notation on minimal models.

Definition 4 ($\mathfrak{I} \leq \mathfrak{I}'$). *Let P be a predicate symbol and let $\mathfrak{I}$ and $\mathfrak{I}'$ be $S \cup \{P\}$-interpretations. Then $\mathfrak{I} \leq^P \mathfrak{I}'$ iff $\mathfrak{I}$ and $\mathfrak{I}'$ agree on all symbols other than P (including variables), and $P^{\mathfrak{I}} \subseteq P^{\mathfrak{I}'}$.*

Definition 5 (P-minimal model). *Let $\mathbb{C}$ be a class of $S \cup \{P\}$-interpretations. Then $\mathfrak{I} \in \mathbb{C}$* is a P-minimal model of $\mathbb{C}$ *iff there is no model $\mathfrak{I}' \in \mathbb{C}$ such that $\mathfrak{I}' \leq^P \mathfrak{I}$ and $\mathfrak{I} \neq \mathfrak{I}'$. If ϕ is an $S \cup \{P\}$-formula, then a P-minimal model of ϕ is a P-minimal model of $Mod(\phi)$, where $Mod(\phi)$ is the class of models of ϕ. We say that a P-minimal model $\mathfrak{I}$ of ϕ is non-empty if $P^{\mathfrak{I}}$ is non-empty.*

Definition 6. *Given an S-interpretation $\mathfrak{I}$, an $S \cup \{P\}$-formula ϕ, and a cardinal α, we say that ϕ has α P-minimal models with respect to $\mathfrak{I}$ if there are α distinct expansions $(\mathfrak{I}, \mathbf{P})$ of $\mathfrak{I}$ which are minimal models of ϕ.*

The relation $\leq^P$ defined on interpretations generalizes the corresponding relation already defined on structures as in [7].

One way to immediately extend MIN(FO) would be allowing the application of the MIN operator to any formula which has a minimum predicate with respect to every S-interpretation (not only extended PIA-conditions). Unfortunately this will lead us to an undecidable language, since it is undecidable whether a formula has only one minimal predicate. To see this, let ϕ be an S-formula and $\psi = \phi \wedge \forall x(P(x))$, where P is a new predicate symbol not occurring in ϕ. Obviously, every model of ψ is a P-minimal model of ψ. If $\mathfrak{I}$ is a model of ϕ, then there exists exactly one expansion $(\mathfrak{I}, \mathbf{P})$ of $\mathfrak{I}$ which is a P-minimal model of ψ. Therefore, ψ has only one P-minimal model with respect to every S-interpretation iff ϕ is valid. Since it is undecidable whether a formula is valid or not, it is undecidable whether a formula like ψ has only one minimal model (with respect to every interpretation).

In face of this problem, we chose to extend MIN(FO) by allowing the application of the MIN operator to any formula. In contrast with PIA-conditions, a generic formula could have more than one minimal model. In order to illustrate this, consider the following example.

Example 1. Let $S = \{c_a, c_b\}$ be a symbol set and $\mathfrak{I}$ be the S-interpretation with domain $A = \{a, b\}$ and such that $c_a^{\mathfrak{I}} = a$ and $c_b^{\mathfrak{I}} = b$. Let ϕ be the formula $P(c_a) \vee P(c_b)$, where P is a predicate variable. There are two possible expansions of $\mathfrak{I}$ which are minimal models of ϕ: an interpretation $\mathfrak{I}'$ with $P^{\mathfrak{I}'} = \{a\}$ and another with $P^{\mathfrak{I}'} = \{b\}$.

How could we interpret the defined predicate $[MINP \bullet \phi]$? The solution is to make some operation on all minimal predicates of ϕ. The most intuitive is to take the intersection or the union of all minimal predicates of ϕ. This will lead us to two different logics described below.

Definition 7 (The U-MIN Logic). *The U-MIN logic extends the first-order logic adding the following rule to the calculus of first-order formulas: if ϕ is a U-MIN formula, P is a predicate variable with arity k and $\overline{t}$ is a tuple of terms of the same length as the arity of P, then $[MIN^u P \bullet \phi](\overline{t})$ is a U-MIN formula. If $\mathfrak{I}$ is an interpretation, then $\mathfrak{I} \models [MIN^u P \bullet \phi](\overline{t})$ iff there is a P-minimal model $(\mathfrak{I}, \mathbf{P})$ of ϕ such that $\overline{t}^{\mathfrak{I}} \in \mathbf{P}$. If there is no such P-minimal model, then $\mathfrak{I} \not\models [MIN^u P \bullet \phi](\overline{t})$.*

The same can be done taking the intersection instead of union.

Definition 8 (The I-MIN Logic). *The I-MIN logic extends the first-order logic adding the following rule to the calculus of first-order formulas: if ϕ is a I-MIN formula, P is a predicate variable with arity k and $\overline{t}$ is a tuple of terms of*

the same length as the arity of P, then $[MIN^i P \bullet \phi](\overline{t})$ is a I-MIN formula. If $\mathfrak{I}$ is an interpretation, then $\mathfrak{I} \models [MIN^i P \bullet \phi](\overline{t})$ iff exists a P-minimal model of ϕ and for all P-minimal model $(\mathfrak{I}, \mathbf{P})$ of ϕ, $\overline{t}^{\mathfrak{I}} \in \mathbf{P}$. If there is no such P-minimal model, then $\mathfrak{I} \not\models [MIN^i P \bullet \phi](\overline{t})$.

Now, we will show that U-MIN and I-MIN are equivalent in expressive power. For it is sufficient to see that for every formula ϕ of U-MIN there is a formula ϕ' of I-MIN with the same models of ϕ, and vice-versa. We will proceed by induction on the structure of ϕ, the difficult case being $\phi = [MIN^i P \bullet \psi](\overline{t})$. By inductive hypothesis, there is a U-MIN formula ψ' equivalent to ψ. Observe that, as they are equivalent, both ψ and ψ' have the same P-minimal models. Let $\phi' = \exists \overline{y}[MIN^u P \bullet \psi'](\overline{y}) \wedge \neg \exists \overline{z}[MIN^u P \bullet \psi' \wedge \neg P(\overline{t})](\overline{z})$. An interpretation $\mathfrak{I} \models \phi'$ iff ψ' has a non-empty P-minimal model $(\mathfrak{I}, \mathbf{P})$ (and hence that any P-minimal model $(\mathfrak{I}, \mathbf{P})$ of ψ' is non-empty), and that there is no P-minimal model $(\mathfrak{I}, \mathbf{P'})$ of $\psi' \wedge \neg P(\overline{t})$. As every P-minimal model of $\psi' \wedge \neg P(\overline{t})$ is a P-minimal model of ψ', this means there is no P-minimal model $(\mathfrak{I}, \mathbf{P'})$ of ψ' such that $\overline{t}^{\mathfrak{I}} \notin \mathbf{P'}$, that is, for all P-minimal model $(\mathfrak{I}, \mathbf{P'})$ of ψ', $\overline{t}^{\mathfrak{I}} \in \mathbf{P'}$. We get $\mathfrak{I} \models \phi$ iff $\mathfrak{I} \models \phi'$.

For the converse, let ϕ be now the U-MIN formula $[MIN^u P \bullet \psi](\overline{t})$. Again, by inductive hypothesis, there is an I-MIN formula ψ' equivalent to ψ. As above, ψ and ψ' have the same P-minimal models. Let $\phi' = \exists \overline{y}[MIN^i P' \bullet [MIN^i P \bullet P \subseteq P' \wedge \psi'](\overline{t})](\overline{y})$, where P' does not occur in ψ'. The predicate P' in the subformula $[MIN^i P \bullet P \subseteq P' \wedge \psi'](\overline{t})$ plays the role of selecting the P-minimal models of ψ' to which $\overline{t}$ belongs, in a way that we can take the intersection of only such P-minimal predicates and verify whether there is some element y in it. If there is some P-minimal model of ψ' to which $\overline{t}$ belongs, $\overline{t}$ will be such y. Using Definitions 7 and 8, we have that an interpretation $\mathfrak{I} \models \phi'$ iff there is a P-minimal model $(\mathfrak{I}, \mathbf{P})$ of ψ' such that $\overline{t}^{\mathfrak{I}} \in \mathbf{P}$ iff $\mathfrak{I} \models \phi$. Then we have proved:

Theorem 3. *The logics I-MIN and U-MIN are equal in expressive power.*

Since I-MIN and U-MIN are proved to be equivalent, we will eliminate the superscripts ^{i} and ^{u} from the MIN operator and assume the U-MIN semantics to evaluate MIN formulas. Hereafter, U-MIN is just MIN. We will occasionally use the superscripts ^{i} and ^{u} whenever necessary.

4 The Expressive Power of MIN

In this section, we will better characterize the logic MIN. We will first introduce the notion of simultaneous minimizing and show this change does not increase the expressive power of MIN. After that, we will see that MIN is equivalent to second-order logic in expressive power, that is, if ϕ is a second-order formula, there is a MIN formula ϕ' which has the same models of ϕ, and vice-versa.

In [13], it is suggested to extend MIN(FO) by allowing simultaneous minimization of several predicates. We will make this notion precise for MIN. First, we will generalize the definition of the relation $\leq^P$ to the case where we have a tuple of predicates.

Definition 9 ($\mathfrak{I} \leq^{\overline{P}} \mathfrak{I}'$)**.** *Let $\overline{P} = P_1, \ldots, P_n$ be a tuple of predicate symbols, and $\mathfrak{I}$ and $\mathfrak{I}'$ be $S \cup \{\overline{P}\}$ interpretations. Then $\mathfrak{I} \leq^{\overline{P}} \mathfrak{I}'$ iff $\mathfrak{I}$ and $\mathfrak{I}'$ agree on all symbols and variables other than P_i, $1 \leq i \leq n$, and $P_i^{\mathfrak{I}} \subseteq P_i^{\mathfrak{I}'}$, $1 \leq i \leq n$. The notion of $\overline{P}$-minimal model is defined analogously as in Definition 5.*

Observe that the relation $\leq^{\overline{P}}$ is related to Parallel Circumscription (for a definition of Parallel Circumscription see [7]). Next, we will define the logic Si-MIN of simultaneous predicate minimizing.

Definition 10 (The Si-MIN Logic). *The Si-MIN logic extends the first-order logic adding the following rule to the calculus of first-order formulas: If $\overline{P} = P_1, \ldots, P_n$ is a tuple of predicate symbols with arities $k_1, \ldots, k_n$ respectively, $\overline{t}$ is a tuple of terms with the same length as the arity of some P_i, $1 \leq i \leq n$, and ϕ is a Si-MIN formula, then $[MINP_i.\overline{P} \bullet \phi](\overline{t})$ is a Si-MIN formula. If $\mathfrak{I}$ is an interpretation, then $\mathfrak{I} \models [MINP_i.\overline{P} \bullet \phi](\overline{t})$ iff there is a $\overline{P}$-minimal model $(\mathfrak{I}, \overline{\mathbf{P}})$ of ϕ such that $\overline{t}^{\mathfrak{I}} \in \mathbf{P}_i$.*

Simultaneous minimizing does not increase the expressive power of MIN. Obviously MIN is contained in Si-MIN. For the converse we need to find, for each Si-MIN formula ϕ, a MIN formula ϕ' equivalent to ϕ. To do that, we will show how to transform a simultaneous minimization into a simple one. The following lemmas will allow us to augment the arity of minimized predicates.

Lemma 1. *Let P and P' be predicate symbols of arity k and $k+1$, respectively. Let $\mathfrak{I}$ be an S-interpretation, $\phi(P)$ an $S \cup \{P\}$ Si-MIN formula, y a variable such that $y^{\mathfrak{I}} = a$, and $\mathbf{P}$ a predicate of the same arity of P. Then $(\mathfrak{I}, \mathbf{P}) \models \phi(P)$ iff $(\mathfrak{I}, \mathbf{P} \times \{a\}) \models \phi(P'\_y)$, where $\phi(P'\_y)$ is obtained from $\phi(P)$ by replacing $P(\overline{t})$ by $P'(\overline{t}, y)$.*

Proof. By induction on $\phi(P)$. □

From this, we get:

Corollary 1. *Let $\overline{P} = P_1, \ldots, P_n$ be a tuple of predicate symbols of arities $k_1, \ldots, k_n$, P_i' a predicate symbol of arity $k_i + 1$, $\phi(\overline{P})$ a Si-MIN formula, y a variable, $\mathfrak{I}$ an interpretation such that $y^{\mathfrak{I}} = a$, and $\overline{\mathbf{P}}$ a tuple of predicates on the universe of $\mathfrak{I}$ with the same arities of $\overline{P}$. Then $(\mathfrak{I}, \overline{\mathbf{P}})$ is a $\overline{P}$-minimal model of $\phi(\overline{P})$ iff $(\mathfrak{I}, \mathbf{P}_1, \ldots, \mathbf{P}_i \times \{a\}, \ldots, \mathbf{P}_n)$ is a $P_1, \ldots, P_i', \ldots, P_n$-minimal model of $\phi(P_1, \ldots, P_i'\_y, \ldots, P_n)$, where $\phi(P_1, \ldots, P_i'\_y, \ldots, P_n)$ is obtained from $\phi(\overline{P})$ by replacing $P_i(\overline{t})$ by $P_i'(\overline{t}, y)$.*

It follows that:

Lemma 2. *Let $\overline{P} = P_1, \ldots, P_n$ be a tuple of predicate symbols where each P_i has arity k_i, and let P_j' be a predicate symbol of arity k_j+1. The formula $[MINP_i.\overline{P} \bullet \phi(\overline{P})](\overline{t})$ is equivalent to*

$$[MINP_i.P_1, \ldots, P_j', \ldots, P_n \bullet \phi(P_1, \ldots, P_j'\_y, \ldots, P_n)](\overline{t}),$$

if $i \neq j$, and to

$$[MINP_j.P_1, \ldots, P'_j, \ldots, P_n \bullet \phi(P_1, \ldots, P'_j\_y, \ldots, P_n)](\overline{t}, y)$$

otherwise.

Proof. Let $\mathfrak{I}$ be an interpretation and $a = y^{\mathfrak{I}}$. If $i = j$, then $\mathfrak{I} \models [MINP_i.\overline{P} \bullet \phi(\overline{P})](\overline{t})$ iff there is a tuple of predicates $\overline{\mathbf{P}}$ such that $(\mathfrak{I}, \overline{\mathbf{P}})$ is a $\overline{P}$-minimal model of $\phi(\overline{P})$ and $\overline{t}^{\mathfrak{I}} \in \mathbf{P}_j$ iff, by Corollary 1, $(\mathfrak{I}, \mathbf{P}_1, \ldots, \mathbf{P}_j \times \{a\}, \ldots, \mathbf{P}_n)$ is a $P_1, \ldots, P'_j, \ldots, P_n$-minimal model of $\phi(P_1, \ldots, P'_j\_y, \ldots, P_n)$ and $(\overline{t}^{\mathfrak{I}}, a) \in \mathbf{P}_j \times \{a\}$ iff $\mathfrak{I} \models [MINP'_j.P_1, \ldots, P'_j, \ldots, P_n \bullet \phi(P_1, \ldots, P'_j\_y, \ldots, P_n)](\overline{t}, y)$. If $i \neq j$, the proof is analogous. □

Iterating Lemma 2 we have:

Corollary 2. *Every Si-MIN formula ϕ is equivalent to a Si-MIN formula ϕ' where all minimized predicates in each MIN operator have the same arity.*

So, without loss of generality we can assume that all minimized predicates in a minimization like $[MINP_i.\overline{P} \bullet \psi](\overline{t})$ have the same arity, say k. After that, we codify the tuple of k-ary predicates $\overline{P}$ into a single $k+1$-ary predicate P'.

Lemma 3. *Let $\overline{P} = P_1, \ldots, P_n$ be a tuple of predicate variables with the same arity k, P' a predicate symbol of arity $k+1$, $\phi(\overline{P})$ a Si-MIN formula, $x_1, \ldots, x_n$ a tuple of variables not occurring in $\phi(\overline{P})$, $\mathfrak{I}$ an interpretation of cardinality greater than or equal to n and such that $x_i^{\mathfrak{I}} = a_i$ and $a_i \neq a_j$, $1 \leq i \neq j \leq n$, and $\overline{\mathbf{P}} = \mathbf{P}_1, \ldots, \mathbf{P}_n$ a tuple of predicates with the same arity k. Then $(\mathfrak{I}, \overline{\mathbf{P}}) \models \phi(\overline{P})$ iff $(\mathfrak{I}, \bigcup_{1 \leq i \leq n}(\mathbf{P}_i \times \{a_i\})) \models \phi(P_1\_x_1, \ldots, P_n\_x_n)$, where $\phi(P_1\_x_1, \ldots, P_n\_x_n)$ is obtained from $\phi(\overline{P})$ by replacing $P_i(\overline{t})$ by $P_i(\overline{t}, x_i)$, $1 \leq i \leq n$.*

Proof. By induction on $\phi(\overline{P})$. □

Here, we point out the following fact. As the symbols occurring in a Si-MIN formula ϕ form a finite symbol set S, the number of S-structures of cardinality less then or equal to l, for a given natural number l, is finite (up to isomorphism). Moreover, since the number of free variables occurring in ϕ is finite, the class of models of ϕ of cardinality less than or equal to l is first-order definable, that is, there is a first-order formula $\phi^{\leq l}$ whose models are exactly the models of ϕ with cardinality less than or equal to l. Now, we are able to prove the following lemma.

Lemma 4. *For each Si-MIN formula ϕ, there is a MIN formula ϕ' equivalent to ϕ.*

Proof. By induction on ϕ. The difficult case is $\phi = [MINP_i.\overline{P} \bullet \psi(\overline{P})](\overline{t})$. By Corollary 2, we can assume that each predicate in $\overline{P}$ has the same arity k. By inductive hypothesis, there is a MIN formula $\psi'(\overline{P})$ equivalent to $\psi(\overline{P})$. Let ϕ' be the formula

$$\phi^{\leq n-1} \vee \exists x_1 \ldots \exists x_n (\bigwedge_{1 \leq l \neq j \leq n} (x_l \neq x_j) \wedge$$

$$[MINP' \bullet \psi'(P'\_x_1, \ldots, P'\_x_n)](\overline{t}, x_i)).$$

If $\mathfrak{I}$ is an interpretation with cardinality less than n, then $\mathfrak{I} \models \phi$ iff $\mathfrak{I} \models \phi^{\leq n-1}$. If the cardinality of $\mathfrak{I}$ is greater than or equal to n, then $\mathfrak{I}$ models ϕ iff there is a $\overline{P}$-minimal model $(\mathfrak{I}, \overline{\mathbf{P}})$ of ψ such that $\overline{t}^{\mathfrak{I}} \in \mathbf{P}_i$ iff, by Lemma 3, there exist n distinct elements $a_1, \ldots, a_n$ in the universe of $\mathfrak{I}$, and a $k+1$-ary predicate $\mathbf{P'}$ such that $(\mathfrak{I}, \mathbf{P'})$ is a P'-minimal predicate of $\psi(\overline{P})$ and $(\overline{t}^{\mathfrak{I}}, a_i) \in \mathbf{P'}$ iff

$$\mathfrak{I} \models \exists x_1 \ldots \exists x_n (\bigwedge_{1 \leq i \neq j \leq n} (x_i \neq x_j) \wedge [MINP' \bullet \psi'(P'\_x_1, \ldots, P'\_x_n)](\overline{t} x_i)).$$

Thus, ϕ is equivalent to ϕ'. □

The expressive equivalence of MIN and Si-MIN follows from Lemma 4.

Theorem 4. *MIN and Si-MIN are equal in expressive power.*

In fact, it can be proved that MIN has the same expressive power of second-order logic, that is, if ϕ is a second-order formula, there is a MIN formula ϕ' which has the same models of ϕ and vice-versa. The technique is the same used in [2] to eliminate fixed predicates from a circumscription. The proof is by induction on ϕ, the difficult case being $\phi = \exists X \psi(X)$. By inductive hypothesis, there is ψ' in MIN which is equivalent to ψ. Let $\phi' = \psi'(\emptyset) \vee \exists \overline{y} [MINP.P, P' \bullet (P = \neg P' \wedge \psi'(P))](\overline{y})$. By using Definitions 9 and 10, we can show ϕ and ϕ' to be equivalent. On the other hand, if $\phi = [MINP \bullet \psi(P)](\overline{y})$, then $\phi' = \exists X(Circ[\psi'(X); X] \wedge X(\overline{y}))$ is equivalent to ϕ, where $Circ[\psi'(X); X]$ is defined following [7] as: $\psi'(X) \wedge \neg \exists Y [\psi'(Y) \wedge Y \subsetneq X]$. Finally, we have proved that:

Theorem 5. *MIN is equivalent to second-order logic in expressive power.*

5 Conclusions

The concept of minimization is widely used in several areas of Computer Science. We pointed out several examples within the scope of Programming Languages, Theory of Computation and Artificial Intelligence. However, if we want to properly formalize the notion of predicate minimization, we know that it is impossible within first-order logic. Nevertheless, there are deductive, monotonic logics more suitable to formalize minimization, as the logic MIN(FO) defined in [13]. A minimal predicate P is defined in MIN(FO) as satisfying a certain first-order formula $\phi(P)$, a PIA-condition.

The logic MIN is here proposed as a generalization of MIN(FO). While the definition of a minimal predicate P results in the minimum one in MIN(FO), the extent of a minimal predicate P is not necessarily unique in MIN. As in MIN(FO), we can iterate predicate minimization in MIN, but the formula $\phi(P)$ need not to follow any specific syntactic pattern as in MIN(FO). We saw that the two possibilities of extending MIN(FO), the U-MIN and I-MIN logics, are equivalent in expressive power by showing that the operators MIN^u and MIN^i

are interdefinable. We also explored the notion of simultaneous minimizing and proved that it does not increase the expressiveness of MIN.

Minimizing a predicate is really a powerful operation. In fact, we established that MIN logic and second-order logic are equally strong. As future work, we want to investigate some fragments of MIN, which are fragments of second-order logic too, obtained by restricting the number of nested MIN operators in a formula. This leads to a hierarchy of expressiveness on such fragments. We will study this hierarchy and relate it with the $\Sigma^1_n, \Pi^1_n, \Delta^1_n$-hierarchy of second-order formulas.

As we have seen, we have used the notion of minimality already used in AI (e.g. Circumscription) and the study of such expressiveness classes sheds light on the study of the expressiveness of Knowledge Representation Systems.

References

1. M.D.Davis, R.Sigal and E.J.Weyuker. *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, 2nd ed. Morgan Kaufmann, San Diego, 1994.
2. J. de Kleer and K. Konolige. Eliminating the Fixed Predicates from a Circumscription. Artificial Intelligence, 39, 391–398, 1989.
3. H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer-Verlag, New York, NY, 1994.
4. Erich Grädel. Guarded fixed point logics and the monadic theory of countable trees. Theoretical Computer Science, 288(1):129–152, 2002.
5. K. Hrbacek and T.Jech. *Introduction to Set Theory*. 2nd edition. Marcel Dekker, INC., New York, 1984.
6. Leonid Libkin. *Elements of Finite Model Theory*. Springer, Berlin, 2004.
7. V. Lifschitz. Circumscription. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, Handbook of Logic in Artificial Intelligence and Logic Programming — Nonmonotonic Reasoning and Uncertain Reasoning (Volume 3), pages 297–352. Clarendon Press, Oxford, 1994.
8. John W. Lloyd. *Foundations of Logic Programming*. 2nd Edition. Springer, New York, 1987.
9. John L. McCarthy. Circumscription — a form of non-monotonic reasoning. Artif. Intell., 13(1–2):27–39, 1980.
10. John L. McCarthy. Applications of circumscription to formalizing common-sense knowledge. Artif. Intell., 28(1), 1986.
11. D. A. Schmidt. *Denotational Semantics: a methodology for language development.* Allyn and Bacon, INC, Boston, 1986.
12. D. Scott. Domains for Denotational Semantics. LNCS 140: Proc. 9th ICALP, pp.577–613, Springer, Berlin, 1982.
13. J. van Benthem. Minimal Predicates, Fixed-Points, and Definability. J. Symbolic Logic, 70(3):696–712, 2005.

Strong Negation in Well-Founded and Partial Stable Semantics for Logic Programs

Pedro Cabalar[1], Sergei Odintsov[2], and David Pearce[3,⋆]

[1] Corunna University, Corunna, Spain
cabalar@udc.es
[2] Sobolev Institute of Mathematics, Novosibirsk, Russia
odintsov@math.nsc.ru
[3] Universidad Rey Juan Carlos, Madrid, Spain
davidandrew.pearce@urjc.es

Abstract. A formalism called *partial equilibrium logic* (PEL) has recently been proposed as a logical foundation for the well-founded semantics (WFS) of logic programs. In PEL one defines a class of minimal models, called *partial equilibrium* models, in a non-classical logic, HT^2. On logic programs partial equilibrium models coincide with Przymusinski's partial stable (p-stable) models, so that PEL can be seen as a way to extend WFS and p-stable semantics to arbitrary propositional theories. We study several extensions of PEL with strong negation and compare these with previous systems extending WFS with explicit negation, notably WSFX [10] and p-stable models with "classical" negation [11].

1 Introduction

The *well-founded semantics* (WFS) of [13] provides one of the most established approaches to logic programming, and the well-known implementation XSB-Prolog[1] is extensively used in AI problem solving. Closely related to WFS is the semantics of *partial stable models* due to Przymusinski [11]. Partial stable (henceforth p-stable) models provide a natural generalisation of stable models [5] to a multi-valued setting and on normal logic programs capture the well-founded model as a special (minimal model) case.

Stable, p-stable and well-founded semantics have all been extended with a second negation operator representing explicit falsity. In the first two cases this was originally called *classical* negation [6,11], in the latter case *explicit* negation [10]. Nowadays the terms "strong" and "explicit" are often used interchangeably to denote this second form of negation.

The term *strong* negation has its roots in logic and refers to the concept of constructible falsity introduced by Nelson [8] and later presented in the form of an axiomatic system by Vorob'ev [14,15]. In the case of stable models or answer sets it is appropriate to label the second negation "strong" because these models

⋆ This research was partially supported by CICyT project TIC-2003-9001-C02.
[1] See http://www.cs.sunysb.edu/∼sbprolog/xsb-page.html

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 592–601, 2006.

correspond in an exact fashion to minimal models in an extension of Nelson's logic, sometimes called $\mathbf{N_5}$ (standing for 5 truth-values). A nonmonotonic extension of $\mathbf{N_5}$, called *equilibrium logic*, yields a foundation for answer set semantics as well as a means to extend the syntax of answer set programs to arbitrary propositional theories [9,7,4]. Another, equivalent way to view $\mathbf{N_5}$ is to take the superintuitionistic logic of *here-and-there*, (HT), and form its least extension by strong negation satisfying the Vorob'ev axioms.

These logical foundations for answer set programs with strong negation have been extensively studied in the literature. Until recently there was no comparable foundational study for well-founded and p-stable semantics; an underlying, monotonic base logic was missing. Recently this situation has changed with the development of *partial equilibrium logic* (PEL), a nonmonotonic formalism proposed in [18] as a logical foundation for well-founded and p-stable semantics. In PEL one defines a class of minimal models, called *partial equilibrium* (p-equilibrium) models, inside a non-classical logic HT^2. It is shown in [18,19] that, on normal and disjunctive logic programs, p-equilibrium models coincide with p-stable models, so that PEL can be seen as a way to extend WFS and p-stable semantics to arbitrary propositional theories.

In this paper we study the result of adding strong negation '$\sim$' to HT^2 and hence to partial equilibrium logic. It is useful to work with a conservative extension HT^2_u of HT^2 formed by adding a constant 'u' for 'undefined'. We consider a basic logical system $HT^{2\sim}_u$ and two different extensions of it. All three are conservative extensions of HT^2_u. The base system is fully paraconsistent in the sense that a literal and its contrary may be true in a single model. One extension, HT^2_{sc}, obeys a "semi-consistency" property requiring that a literal and its contrary cannot be true in a single model. Another extension, HT^2_{coh}, obeys a form of the so-called *coherence* principle [10], sometimes formulated by $\sim L \rightarrow \neg L$, for any literal L, where '$\neg$' is default negation. Corresponding to these three underlying logics, one obtains different variants of partial equilibrium logic with strong negation.

While $HT^{2\sim}_u$ can already be considered suitable for handling paraconsistent strong negation in this setting, its extensions relate to semantics proposed for explicit negation in logic programming. Specifically we show that PEL based on HT^2_{sc} captures precisely Przymusinski's partial stable semantics for programs with "classical" negation [11]. Furthermore, by a suitable encoding of program rules into a particular kind of implications, the semantics WSFX of Pereira and Alferes [10] is also interpretable in HT^2_{sc}. Finally, HT^2_{coh} captures the so-called WFS with "strong negation" as defined in [2].

2 Routley Semantics and Strong Negation

The logic HT^2 underlying partial equilibrium logic was defined in [18] as a finite-valued extension of a logic N^* whose semantics is given by direct combination of Kripke semantics for intuitionistic logic and Routley style semantics for weak

negation $\neg$ [12]. Prior to considering the version of HT^2 with strong negation, we first introduce this kind of negation to N^*, leading to a new logic we will denote as $N^{*\sim}$. Formulas of N^* are built-up in the usual way using the logical constants: $\wedge$, $\vee$, $\rightarrow$, $\neg$, standing respectively for conjunction, disjunction, implication and weak negation; $N^{*\sim}$ adds to these strong negation, $\sim$. The rules of inference for N^* and $N^{*\sim}$ are *modus ponens* and the contraposition rule for weak negation

$$\text{(RC)}\ \frac{\alpha \rightarrow \beta}{\neg\beta \rightarrow \neg\alpha}$$

The axioms of N^* ($N^{*\sim}$) are as follows:

1. the axiom schemes of positive logic,
2. weak negation axioms:

   W1. $\neg\alpha \wedge \neg\beta \rightarrow \neg(\alpha \vee \beta)$ **W2.** $\neg(\alpha \wedge \beta) \rightarrow \neg\alpha \vee \neg\beta$
   W3. $\neg(\alpha \rightarrow \alpha) \rightarrow \beta$

3. and for $N^{*\sim}$ one adds the following axiom schemata involving strong negation taken from the calculus of Vorob'ev [14,15] :

   N1. $\sim(\alpha \rightarrow \beta) \leftrightarrow \alpha \wedge \sim\beta$ **N2.** $\sim(\alpha \wedge \beta) \leftrightarrow \sim\alpha \vee \sim\beta$
   N3. $\sim(\alpha \vee \beta) \leftrightarrow \sim\alpha \wedge \sim\beta$ **N4.** $\sim\sim\alpha \leftrightarrow \alpha$
   N5. $\sim\neg\alpha \leftrightarrow \alpha$

We introduce a notion of frame in the style of Routley [12].

Definition 1 ($N^{*\sim}$ model). *A Routley or $N^{*\sim}$* frame *is a triple $\langle W, \leq, * \rangle$, where W is a set, $\leq$ a partial order on W and $* : W \longrightarrow W$ is such that $x \leq y$ iff $y^* \leq x^*$. An $N^{*\sim}$* model *$\mathcal{M} = \langle W, \leq, *, V^+, V^- \rangle$ is an $N^{*\sim}$ frame $\langle W, \leq, * \rangle$ together with two valuations $V^+, V^- : At \times W \longrightarrow \{0, 1\}$ satisfying the condition:*

$$V^{+(-)}(p, u) = 1 \ \ \& \ \ u \leq w \ \ \Rightarrow \ \ V^{+(-)}(p, w) = 1 \tag{1}$$

Valuations extend to all formulas in the following way:

- $V^+(\varphi \wedge \psi, w) = 1$ iff $V^+(\varphi, w) = V^+(\psi, w) = 1$
- $V^+(\varphi \vee \psi, w) = 1$ iff $V^+(\varphi, w) = 1$ or $V^+(\psi, w) = 1$
- $V^+(\varphi \rightarrow \psi, w) = 1$ iff for every w' such that $w \leq w'$, $V^+(\varphi, w') = 1 \Rightarrow V^+(\psi, w') = 1$
- $V^+(\neg\varphi, w) = 1$ iff $V^+(\varphi, w^*) = 0$
- $V^+(\sim\varphi, w) = 1$ iff $V^-(\varphi, w) = 1$
- $V^-(\varphi \wedge \psi, w) = 1$ iff $V^-(\varphi, w) = 1$ or $V^-(\psi, w) = 1$
- $V^-(\varphi \vee \psi, w) = 1$ iff $V^-(\varphi, w) = V^-(\psi, w) = 1$
- $V^-(\varphi \rightarrow \psi, w) = 1$ iff $V^+(\varphi, w) = 1$ and $V^-(\psi, w) = 1$
- $V^-(\neg\varphi, w) = 1$ iff $V^+(\varphi, w) = 1$
- $V^-(\sim\varphi, w) = 1$ iff $V^+(\varphi, w) = 1$

The only difference between $N^{*\sim}$ models and Routley models for the logic N^* as defined in [18] is the presence of two valuations V^+ and V^- and valuation V^+ is defined in exactly the same way as the only valuation V of a Routley model. We read $V^+(\varphi, w) = 1$ as "φ is verified at w" and $V^-(\varphi, w) = 1$ as "φ is falsified at w". A proposition φ is said to be *true* in an $N^{*\sim}$ model $\mathcal{M} = \langle W, \leq, R, V^+, V^- \rangle$, and we write $\mathcal{M} \models \varphi$, if $V^+(\varphi, v) = 1$, for all $v \in W$. A formula φ is *valid*, in symbols $\models \varphi$, if it is true in every $N^{*\sim}$ model. It is easy to prove by induction that condition (1) above holds for any formula φ, ie, $V^{+(-)}(\varphi, u) = 1 \ \ \& \ \ u \leq w \Rightarrow V^{+(-)}(\varphi, w) = 1$. Moreover $N^{*\sim}$ is complete for this semantics in the sense that a formula is valid iff it is a theorem of N. Completeness proof for $N^{*\sim}$ can be obtained via an easy modification of the canonical model method used in [18] to prove the completeness of N^* wrt Routley models.

Note that axiom (**W3**) allows to define an intuitionistic negation '$-$' in $N^{*\sim}$. Fix some propositional variable p_0 and put $\bot := \neg(p_0 \to p_0)$ and $-\alpha := \alpha \to \bot$, then the $\langle \vee, \wedge, \to, - \rangle$-fragment of $N^{*\sim}$ coincides with intuitionistic logic.

Verification and falsification for $\vee$, $\wedge$, $\to$ and $\sim$ was defined in exactly the same way as for paraconsistent Nelson's logic N^- [17], at the same time verification conditions for $\vee$, $\wedge$, $\to$ and $\neg$ are the same as validity conditions for these connective in the logic N^* [18], whence

Proposition 1. *The logic $N^{*\sim}$ is a conservative extension of both N^- and N^*.*

We say that a formula is in negative normal form (*nnf*) if it contains strong negation only in front of atoms. The strong negation axioms allow one to move strong negation through all other connectives, therefore, we have

Proposition 2. *For any formula φ, there is a formula ψ in* nnf *such that $N^{*\sim} \vdash \varphi \leftrightarrow \psi$.*

Assume that we have chosen one or another procedure for reducing a formula φ to its *nnf* $\overline{\varphi}$. Now we assign to each atom $p \in At$ the new atom p' and to each formula φ the formula φ' obtained by replacing in $\overline{\varphi}$ each subformula $\sim p$ by p'. In this way we obtain an embedding of $N^{*\sim}$ into N^*. This method for eliminating strong negation was discovered by Vorob'ev [14,15].

Proposition 3 (Vorob'ev reduction). *$N^{*\sim} \vdash \varphi$ iff $N^* \vdash \varphi'$.*

3 HT^2 with Strong Negation

Given a semantic characterisation of some N^* extension determined via a class of N^* frames, the least extension of this logic with strong negation can be defined by considering $N^{*\sim}$ models over the same class of frames. In this way we obtain the definition of $HT^{2\sim}$ from the logic HT^2. We first introduce briefly the latter.

The logical constants of HT^2 are $\wedge$, $\vee$, $\to$, $\neg$. Intuitionistic negation '$-$' is definable as $-\alpha := \alpha \to \neg(p_0 \to p_0)$. The axioms and inference rules are those of N^* together with the schemata:

W4. $-\alpha \vee - - \alpha$
W5. $-\alpha \vee (\alpha \rightarrow (\beta \vee (\beta \rightarrow (\gamma \vee -\gamma))))$
W6. $\bigwedge_{i=0}^{2}((\alpha_i \rightarrow \bigvee_{j\neq i} \alpha_j) \rightarrow \bigvee_{j\neq i} \alpha_j) \rightarrow \bigvee_{i=0}^{2} \alpha_i$
W7. $\alpha \rightarrow \neg\neg\alpha$
W8. $\alpha \wedge \neg\alpha \rightarrow \neg\beta \vee \neg\neg\beta$
W9. $\neg\alpha \wedge \neg(\alpha \rightarrow \beta) \rightarrow \neg\neg\alpha$
W10. $\neg\neg\alpha \vee \neg\neg\beta \vee \neg(\alpha \rightarrow \beta) \vee \neg\neg(\alpha \rightarrow \beta)$
W11. $\neg\neg\alpha \wedge \neg\neg\beta \rightarrow (\alpha \rightarrow \beta) \vee (\beta \rightarrow \alpha)$

and the rule (EC) $\frac{\alpha \vee (\beta \wedge \neg\beta)}{\alpha}$

HT^2 and $HT^{2\sim}$ models are based on the same notion of frame:

Definition 2 ($HT^{2\sim}$, HT^2 frame). *An $HT^{2\sim}$ (HT^2)* frame *is an $N^{*\sim}$ frame $\mathcal{M} = \langle W, \leq, * \rangle$ such that (i) W comprises 4 worlds denoted by h, h', t, t', (ii) $\leq$ is a partial ordering on W satisfying $h \leq t$, $h \leq h'$, $h' \leq t'$ and $t \leq t'$, (iii) the $*$-operation is given by $h^* = t^* = t'$, $h'^* = t'^* = t$.*

Definition 3 ($HT^{2\sim}$, HT^2 model). *An $HT^{2\sim}$* model *$\mathcal{M} = \langle W, \leq, *, V^+, V^- \rangle$ is an $HT^{2\sim}$ frame together with $N^{*\sim}$ valuations $V^+, V^- : At \times W \longrightarrow \{0, 1\}$. An HT^2 model is like an $HT^{2\sim}$ model but with a single valuation V corresponding to V^+ and extended in that way to all N^* formulas.*

In [18] it was shown that HT^2 is complete for the class of HT^2 models.

Proposition 4. *The Vorob'ev embedding also holds for $HT^{2\sim}$, that is: $HT^{2\sim} \vdash \varphi$ iff $HT^2 \vdash \varphi'$.*

With the help of the Vorob'ev reduction one can easily pass from the axioms of HT^2 to obtain those of $HT^{2\sim}$.

Theorem 1. $HT^{2\sim} = HT^2 + \{\boldsymbol{N1}, \ldots, \boldsymbol{N5}\}$.

Proof. HT^2 axioms are $\sim$-free and only valuation V^+ is involved in checking their validity. Since V^+ coincides with valuation V of HT^2 models all these axioms hold in $HT^{2\sim}$-models too. Vorob'ev axioms hold in all $N^{*\sim}$ models, in particular, in $HT^{2\sim}$. To prove completeness let us take some $\varphi \notin HT^{2\sim}$. By Proposition 4, $\varphi' \notin HT^2$. Let $\mathcal{M}' = \langle W, \leq, *, V \rangle$ be a counter model for φ'. Defining $V^+(p, w) = V(p, w)$ and $V^-(p, w) = V(p', w)$ we obtain a countermodel $\mathcal{M} = \langle W, \leq, *, V^+, V^- \rangle$ for φ. □

Corollary 1. *$HT^{2\sim}$ is the least $N^{*\sim}$-extension with the property that it is a conservative extension of HT^2.*

The logic $HT^{2\sim}$ is suitable to handle paraconsistent strong negation. Before defining logics suitable for explosive versions of strong negation with or without a coherence principle we have to introduce an additional constant into HT^2. Let u be a new constant symbol that will stand for "undefinedness."

Definition 4. *An HT^2_u-model is an HT^2 model, where the constant u is interpreted so that $V(u, h) = V(u, t) = 0$ and $V(u, h') = V(u, t') = 1$.*

Denote by HT^2_u the logic determined by the class of HT^2_u-models.

Theorem 2. *HT^2_u is a conservative extension of HT^2. Moreover, $HT^2_u = HT^2 + \{u \leftrightarrow \neg u\}$.*

The proof is based on an easy observation that formula $p \leftrightarrow \neg p$ is true in an HT^2 model iff p is true at h' and t' and false at h and t. We define the next three conservative extensions of HT^2_u with strong negation:

$$\begin{aligned} HT^{2\sim}_u &:= HT^2_u + \{\mathbf{N1}, ..., \mathbf{N5}\}, \\ HT^2_{sc} &:= HT^{2\sim}_u + \{p \wedge \sim p \rightarrow u\}, \\ HT^2_{coh} &:= HT^{2\sim}_u + \{p \rightarrow \neg \sim p \vee u, \sim p \rightarrow \neg p \vee u\}. \end{aligned}$$

As above, $HT^{2\sim}_u$ is the least conservative extension of HT^2_u with strong negation. Models of the other logics are characterised as follows.

Proposition 5. *Let $\mathcal{M} = \langle W, \leq, *, V^+, V^- \rangle$ be an $HT^{2\sim}_u$ model.*

(sc) *$\mathcal{M} \models HT^2_{sc}$ iff there is no atom p such that $V^+(p,t) = V^-(p,t) = 1$.*

(coh) *$\mathcal{M} \models HT^2_{coh}$ iff for any atom p,*
$V^+(p,t) = 1 \;\Rightarrow\; V^-(p,t') = 0, \quad V^-(p,t) = 1 \;\Rightarrow\; V^+(p,t') = 0.$

One can see that in the case of HT^2_{sc} a principle of partial or semi-consistency (sc) is satisfied, whereas for HT^2_{coh} the right-hand condition of Item 2 is equivalent to the coherence principle (coh). Although these conditions will be explained in further detail in the next section, it is easy to see that (coh) implies (sc) and so:

Proposition 6. *HT^2_{coh} is stronger than HT^2_{sc}.*

We conclude this section by formulating Vorob'ev reductions for these logics.

Proposition 7. *Let φ be a formula in the language $\{\vee, \wedge, \rightarrow, \neg, \sim, u\}$ and $At(\varphi)$ denote the set of its atoms.*

1. *$HT^{2\sim}_u \vdash \varphi$ iff $HT^2_u \vdash \varphi'$*
2. *$HT^2_{sc} \vdash \varphi$ iff $HT^2_u \vdash (\bigwedge_{p \in At(\varphi)} p \wedge p' \rightarrow u) \;\rightarrow\; \varphi'$.*
3. *$HT^2_{coh} \vdash \varphi$ iff $HT^2_u \vdash (\bigwedge_{p \in At(\varphi)} (p \rightarrow \neg p' \vee u) \wedge (p' \rightarrow \neg p \vee u)) \;\rightarrow\; \varphi'$.*

4 Partial Equilibrium Logic with Strong Negation

At each world in an $HT^{2\sim}$ or $HT^{2\sim}_u$ model $\mathcal{M} = \langle W, \leq, ^*, V^+, V^- \rangle$ a certain set of literals (atoms or strongly negated atoms) is verified. Let us denote by H, H', T, T' the four sets of literals respectively verified at each corresponding world h, h', t, t'. In case $\mathcal{M}$ is an HT^2 model, H, H', T, T' are simply sets of atoms. More succinctly, we can represent $\mathcal{M}$ as the pair $\langle \mathbf{H}, \mathbf{T} \rangle$ so that we group each pair of unprimed/primed worlds as $\mathbf{H} = (H, H')$ and $\mathbf{T} = (T, T')$. Notice that $H \subseteq H'$ and $T \subseteq T'$ by construction of $\mathcal{M}$. Given such pairs $\mathbf{H}, \mathbf{T}$ we define a partial ordering relation by $\mathbf{H} \leq \mathbf{T}$ iff $H \subseteq T$ and $H' \subseteq T'$. Note that if $\langle \mathbf{H}, \mathbf{T} \rangle$ is a model then necessarily $\mathbf{H} \leq \mathbf{T}$. The ordering $\leq$ is extended to a partial ordering $\trianglelefteq$ among models as follows. We set $\langle \mathbf{H}_1, \mathbf{T}_1 \rangle \trianglelefteq \langle \mathbf{H}_2, \mathbf{T}_2 \rangle$ if (i)

$\mathbf{T}_1 = \mathbf{T}_2$; (ii) $\mathbf{H}_1 \leq \mathbf{H}_2$. A model $\langle \mathbf{H}, \mathbf{T} \rangle$ in which $\mathbf{H} = \mathbf{T}$ is said to be *total*. Note that the term *total* model does not refer to the absence of undefined literals. To represent this, we further say that a total partial equilibrium model is *complete* if $\mathbf{T}$ has the form (T, T).

Let Π be a set of formulas and let $\mathcal{M}$ range over HT^2 or $HT^{2\sim}_u$ models as appropriate.

Definition 5 (Partial equilibrium model). *A model $\mathcal{M}$ of Π is said to be a* partial equilibrium *(or p-equilibrium) model of Π if (i) $\mathcal{M}$ is total; (ii) $\mathcal{M}$ is minimal among models of Π under the ordering $\trianglelefteq$.*

In other words a p-equilibrium model of Π has the form $\langle \mathbf{T}, \mathbf{T} \rangle$ and is such that if $\langle \mathbf{H}, \mathbf{T} \rangle$ is any model of Π with $\mathbf{H} \leq \mathbf{T}$, then $\mathbf{H} = \mathbf{T}$. *Partial equilibrium logic* (PEL) is the logic determined by truth in all p-equilibrium models of a theory.

We turn to the relation between PEL and logic programs. A *disjunctive (resp. normal) logic program* is a set of formulas (also called *rules*) of the form

$$a_1 \wedge \ldots \wedge a_m \wedge \neg b_1 \wedge \ldots \wedge \neg b_n \rightarrow c_1 \vee \ldots \vee c_k \tag{2}$$

where $m, n, k \geq 0$ (resp. $m, n, \geq 0, k = 1$), and a, b, c with subscripts range over atoms. A program is called *extended* if a, b, c with subscripts range over *objective literals* (ie, an atom p or its strong negation $\sim p$). For reasons of space we do not repeat here the definition of partial stable (p-stable) models of [11]. A central result of [18,19] is:

Theorem 3 ([19]). *A total HT^2 model $\langle \mathbf{T}, \mathbf{T} \rangle$ is a p-equilibrium model of a disjunctive program Π iff $\mathbf{T}$ is a p-stable model of Π.*

The obvious corresponding variants of PEL with strong negation are obtained by considering $HT^{2\sim}_u$, $HT^{2\sim}_{sc}$ or $HT^{2\sim}_{coh}$ models respectively. Let us now reconsider the conditions given in Proposition 5. We see that models of $HT^{2\sim}_{sc}$ are 'semi'-consistent in that the sets H, T (ie, the "founded" information) do not contain any contrary pairs of literals $p, \sim p$. On the other hand, models of $HT^{2\sim}_{coh}$ have the property that $p \in T \Rightarrow \sim p \notin T'$ and $\sim p \in T \Rightarrow p \notin T'$. Hence in any p-equilibrium model $\mathcal{M}$ over $HT^{2\sim}_{coh}$, we have that $\mathcal{M} \models p \Rightarrow \mathcal{M} \models \neg\sim p$ and $\mathcal{M} \models \sim p \Rightarrow \mathcal{M} \models \neg p$. This is exactly the coherence principle formulated in [2].

The fact that the logics $HT^{2\sim}_u$, $HT^{2\sim}_{sc}$ and $HT^{2\sim}_{coh}$ are reducible to HT^2_u via the Vorob'ev transformations given in Proposition 7 is not only significant for proving completeness theorems. The transformation from φ to φ' is linear and allows us to transfer some key properties from ordinary PEL to the strong negation variants. This applies in particular to the strong equivalence theorem and complexity results. We say that two sets of formulas Γ_1 and Γ_2 are *strongly equivalent*, if for any set of formulas Γ, $\Gamma_1 \cup \Gamma$ and $\Gamma_2 \cup \Gamma$ have the same p-equilibrium models.

Theorem 4 ([18]). *Sets of N^* formulas Γ_1 and Γ_2 are strongly equivalent iff Γ_1 and Γ_2 are equivalent as HT^2_u theories.*

Corollary 2. *Theorem 4 continues to hold for sets of $N^{*\sim}$ formulas where strong equivalence is defined wrt any of the given strong negation extensions of PEL and HT^2_u is replaced by $HT^{2\sim}_u$, $HT^{2\sim}_{sc}$ and $HT^{2\sim}_{coh}$ as appropriate.*

Proposition 8. *The complexity of reasoning tasks in (any of the variants of) PEL with strong negation lies in the same class as that of ordinary PEL. In particular, from [19] it follows that the decision problem for entailment in PEL with strong negation (checking truth in all p-equilibrium models) is Π_2^P-hard.*

Note that as in the case of answer set semantics, the Vorob'ev transformation allows reducing all strong negation variants to ordinary PEL, whose implementation strategies are discussed elsewhere [19].

5 Capturing WFS Variants with a Second Negation

Let us focus now on extended logic programs without disjunction, consisting of rules like (2) where $k = 1$, and where subscripted a, b, c are objective literals. We denote $B^+(r) := \{a_1, \ldots, a_m\}$, $B^-(r) := \{b_1, \ldots, b_m\}$ and $Hd(r) := c_1$. Sometimes $B^+(r)$ (resp. $B^-(r)$) will be used as the conjunction (resp. the disjunction) of their atoms. The whole body is denoted as $B(r) := B^+(r) \wedge \neg B^-(r)$. A program is said to be *definite* if $B^-(r) = \emptyset$ for all its rules. Given a set of objective literals I and a definite rule r, we write $I \models r$ when $Hd(r) \in I$ if $B^+(r) \subseteq I$ and $I \models P$ when $I \models r$ for all $r \in P$. The set I is *consistent* when it does not contain a pair of literals p and $\sim p$. The *reduct* [5] P^I of a normal logic program P wrt interpretation I corresponds to: (i) removing all $r \in P$ with $B^-(r) \cap I \neq \emptyset$; and (ii) removing the default literals from the remaining rules. As P^I is definite, it has a least model usually represented as $\Gamma_P(I)$, or just $\Gamma(I)$ when there is no ambiguity. The *seminormal version* P_s of a program P consists of a rule like $\neg \sim Hd(r) \wedge B(r) \rightarrow Hd(r)$ per each rule $r \in P$. We write Γ_s to stand for Γ_{P_s}, when there is no ambiguity about P. Furthermore, we consider that $\Gamma_s(I)$ is not defined for an inconsistent I.

A *partial stable (p-stable) model* is a pair $I \subseteq J$ satisfying $J = \Gamma(I)$ and $I = \Gamma(J)$. When I is consistent, it is called p-stable model with "*classical negation*" [11]. When (I, J) further satisfies coherence ($L \in I$ implies $\sim L \notin J$) it is called p-stable model with "*strong negation*" [2]. A *WFSX p-stable model* is any pair of sets of literals (I, J), with I consistent and $I \subseteq J$, satisfying $J = \Gamma_s(I)$ and $I = \Gamma(J)$. The *well-founded model* (WFM) is the least-information p-stable model (if one exists) in each of these variants. The WFM is computable by iteration on operator $\Gamma\Gamma$ or, in the case of WFSX, on $\Gamma\Gamma_s$.

Theorem 5. *$\langle \mathbf{T}, \mathbf{T} \rangle$ is an HT^2_{sc} p-equilibrium model of an extended program P iff $\mathbf{T}$ is a classical-negation p-stable model of P.*

Theorem 6. *$\langle \mathbf{T}, \mathbf{T} \rangle$ is an HT^2_{coh} p-equilibrium model of an extended program P iff $\mathbf{T}$ is a strong-negation p-stable model of P.*

Our WFSX encoding translates a program P into P' adding, for each $r \in P$, the pair of implications: $u \wedge \neg \sim Hd(r) \wedge B(r) \rightarrow Hd(r)$ and $B(r) \rightarrow Hd(r) \vee u$.

Lemma 1. *Let $\mathcal{M} = \langle \mathbf{H}, \mathbf{T} \rangle$, with $\mathbf{H} = (H, H')$ and $\mathbf{T} = (T, T')$, and let P be an extended logic program. Then $\mathcal{M} \models P'$ in HT^2_u is equivalent to the four conditions: $H' \models P_s^T$, $H \models P^{T'}$, $T' \models P_s^T$ and $T \models P^{T'}$.*

Theorem 7. *A pair* $\mathbf{T} = (T, T')$ *is a WFSX p-stable model of an extended logic program* P *iff* $\langle \mathbf{T}, \mathbf{T} \rangle$ *is an* HT^2_{sc} *p-equilibrium model of* P'.

Proof (sketch). The left to right direction follows quite directly from Lemma 1. For right to left, if $\langle \mathbf{T}, \mathbf{T} \rangle$ is an HT^2_{sc} p-equilibrium model of P', from Lemma 1 we get $T \models P^{T'}$ and $T' \models P^T_s$, so we only have to prove their minimality. If we have some $H \subset T$, $H \models P^{T'}$, then we can use Lemma 1 to conclude that the interpretation $\langle (H, T'), \mathbf{T} \rangle$ would be model of P' smaller than $\langle \mathbf{T}, \mathbf{T} \rangle$. If we have some $H' \subset T'$, $H' \models P^T_s$, something similar happens applying Lemma 1 on the interpretation $\langle (H' \cap T, H'), \mathbf{T} \rangle$, although in this case, to prove that $H' \cap T \models P^{T'}$ we must use consistency of T. □

6 Concluding Remarks

Partial equilibrium logic (PEL) provides a purely logical characterisation of p-stable models and hence, as argued in [18,19], yields a natural logical foundation for p-stable semantics and WFS. We have shown here how strong negation can be added to PEL while maintaining the same complexity of reasoning and preserving strong equivalence theorems. While the coherence principle ($\sim L \rightarrow \neg L$), much discussed in [3], seems plausible for explosive strong negation, originally [10] strong negation was added to logic programs without this principle. When we deal with strong negation in a paraconsistent setting it seems reasonable to explore different options: logics including coherence as in [1] as well as logics without, as defined here. Indeed, if we do not assume that models are consistent, the fact $\sim p$ should not imply that we can not prove p. In this way we are led to the three conservative extensions of HT^2 with strong negation, viz. $HT^{2\sim}_u$, $HT^{2\sim}_{sc}$ and $HT^{2\sim}_{coh}$, where only the latter satisfies coherence.

We saw that PEL over $HT^{2\sim}_{coh}$ provides an alternative to the semantics WFSX since its p-equilibrium models do not coincide with the p-stable models of [3]. Evidently there are different solutions for adding a second, coherent negation to programs under WFS. There may be a trade-off between obtaining intuitively correct inferences and computationally desirable properties. For example, WSFX over normal programs enjoys computationally useful properties like relevance [3] (not obeyed by PEL in general), while some example programs discussed above seem to lead to less intuitive inferences under WSFX than under PEL. Nevertheless we have seen a precise sense in which WSFX is interpretable in PEL over $HT^{2\sim}_{sc}$; moreover PEL offers a genuinely logical treatment of strong negation, defined over arbitrary formulas. In the case of the p-stable semantics of Przymusinski [11], on the other hand, we can derive precise agreement with the version of PEL defined over the logic $HT^{2\sim}_{sc}$, and therefore obtain a means to extend p-stable semantics to arbitrary propositional theories.

Summarising, we have studied extensions of PEL with strong negation based on natural, logical principles. Capturing WSFX or other specific semantics was not a primary aim. We have provided precise logical characterisations for various extensions using axiomatic systems and we proved Strong Equivalence theorems. As a by-product we showed that some of our extensions are closely related to

previously proposed semantics. In [1] there is a frame-based approach to WSFX and its paraconsistent variant, WSFXp, but the aims and approach seem to be different. [1] does not discuss logic, provide axiomatisations or strong equivalence theorems. For these reasons, and for reasons of space, we have not provided a comparison with the work of [1]; this topic is postponed for the future.

References

1. J. Alcântara, C. Damásio & L. M. Pereira. A Frame-based Characterisation of the Paraconsistent Well-founded Semantics with Explicit Negation. Unpublished draft, available at `http://centria.di.fct.unl.pt/~jfla/publications/`
2. Alferes, J.J., Pereira, L.M.: On logic programs semantics with two kinds of negation. In: Proc. of the 5th Intl. Joint Conf. and Symposium on Logic Programming, MIT Press (1992) 574–588.
3. J.J. Alferes & L.M. Pereira. *Reasoning with Logic Programming*, Springer, 1996.
4. P. Cabalar, D. Pearce & A. Valverde. Reducing propositional Theories in Equilibrium Logic to Logic Programs in *Proc. EPIA 2005*, LNAI 3803, Springer, 2005.
5. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP'88*, pp. 1070–1080, 1988. The MIT Press.
6. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
7. V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, October 2001.
8. D. Nelson. Constructible falsity. *Journal of Symbolic Logic*, 14(2):16–26, 1949.
9. D. Pearce. A new logical characterisation of stable models and answer sets. In *Proc. of NMELP 96*, LNCS 1216, pp. 57–70. Springer, 1997.
10. L. M. Pereira & J. J. Alferes, Well Founded Semantics for Logic Programs with Explicit Negation, in: *European Conference on Artificial Intelligence*, B. Neumann (ed.), John Wiley & Sons 1992, pp.102-106.
11. Przymusinski, T. Stable semantics for disjunctive programs. *New Generation Computing* 9 (1991), 401-424.
12. R. Routley and V. Routley. The Semantics of First Degree Entailment. *Noûs*, 6, 335–359, 1972.
13. A. van Gelder, K.A. Ross, and J.S. Schlipf. Unfounded sets and well-founded semantics for general logic programs. *JACM*, 38(3):620–650, 1991.
14. N. N. Vorob'ev. A constructive propositional calculus with strong negation (in Russian). *Doklady Akademii Nauk SSR*, 85:465–468, 1952.
15. N. N. Vorob'ev. The problem of deducibility in constructive propositional calculus with strong negation (in Russian). *Doklady Akademii Nauk SSR*, 85:689–692, 1952.
16. N. N. Vorob'ev. Constructive propositional calculus with strong negation (in Russian). *Transactions of Steklov's Institute*, 72: 195-227, 1964.
17. H. Wansing. Negation. In: L.Goble(ed.), *The Blackwell Guide to Philosophical Logic*, Basil Blackwell Publishers, Cambridge/MA, 415–436, 2001.
18. Logical Foundations of Well-Founded Semantics. to appear, 2006.
19. Analysing and Extending Well-Founded and Partial Stable Semantics using Partial Equilibrium Logic. Technical Report, Feb. 2006, submitted for publication.

MAT Logic: A Temporal×Modal Logic with Non-deterministic Operators to Deal with Interactive Systems in Communication Technologies

Gabriel Aguilera[1], Alfredo Burrieza[2], Pablo Cordero[1], Inmaculada P. de Guzmán[1], and Emilo Muñoz[1]

[1] Departamento de Matemática Aplicada
[2] Departamento de Filosofía,
Universidad de Málaga. 29071-Spain
gabri@ctima.uma.es, burrieza@uma.es, pcordero@uma.es, guzman@ctima.uma.es, emilio@ctima.uma.es

Abstract. In this paper, the Multi-flow Asynchronous Temporal Logic, called MAT Logic, is presented. MAT Logic is a new temporal×modal logic with non-deterministic operators among time flows as accessibility relations. The main goal of this work has been the design and description of a logic that could be capable of managing communications among systems with not necessarily synchronizable time flows. In order to better understand the design of the logic, an example in the field of communications is given.

1 Introduction

The necessity of the incorporation of non-determinism in computation has been widely discussed. So, for example, in the literature, the concept of non-deterministic automata as a formal model of computation is widely consolidated; in [20] the author presents a discussion about how the study of non-determinism is useful for natural language processing; in [10] the author shows how formal non-deterministic models are useful in describing interactive systems. Another example is designing a circuit or a network: non-determinism characterizes the flexibility allowed in the design [15].

Most works about non-determinism are based on simulation by means of algorithms and deterministic automata. Nonetheless, it is widely accepted that it will be necessary to develop a formal theory that regards non-determinism as inherent to it and the fact that computational logic will play an important role in this development [12].

Thus, on the one hand, modal logics have been proven useful in interactive systems. So, they have been used in multi-agent systems to describe the agent mental state and behaviour [17], or, for example, to reason with social categories, such as obligations [3] and cooperativity [1].

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 602–611, 2006.

On the other hand, temporal logic has been shown as a successful tool for specifying and reasoning with interactive systems and the global behaviour of multi-agent systems. However, it is not capable of reasoning out the intern structure of these systems [8,13,16]. In the literature, there exist several extensions of propositional temporal logic to solve this disadvantage. So, for example, in the case of multi-agent systems, the simplest extension is to consider that all the agents are synchronized [9,16], nevertheless, this is a very strong restriction. Other extensions are obtained via some form of synchronization given by *visibility* or *accessibility* functions. Thus, temporal logics with linear temporal flows in which the visibility functions are bidirectional, that is, the relation among states (in different flows) is *symmetric*, were introduced in [14,18].

In our opinion, a combination of the above approaches, i.e. modal and temporal logics, could be the key to achieve a more comprehensive way to describe interactive and multi-agent systems. Nonetheless, determining which properties of the chosen combinations hold is not an easy task [21]. In the framework of combining this kind of logics, this work presents the Multi-flow Asynchronous Temporal Logic (briefly, MAT Logic). Our main goal has been the design and description of a logic that could be capable of managing communications among systems with time flows not necessarily synchronized. Occasionally, this kind of communication between two time flows can be described by a function. However, on many occasions the type of instant (kind of state of the system) in the image flow is known but not the specific instant, consequently, a function can not be defined. These characteristics, together with the fact that synchronization of the time flows is not required, have led us to represent accessibility among them by means of, on the one hand, *non-deterministic operators* for possible communications and, on the other hand, *execution functions* for effective communications.

The usual way in the literature about temporal×modal logics is to use equivalence relations of accessibility, for example the Kamp-models in [19] and the reasoning about knowledge and time in asynchronous systems, in [11]. However, in [6,7] a new kind of frame was introduced to manage linear time flows connected by accessibility functions instead of using equivalence relations. MAT Logic is a more general framework, because the accessibility is given by non deterministic operators and, as a consequence, can be applied to different situations only by changing the properties required to them. Another characteristic to be considered, is the use of indexed connectives to label time flows that can be reached from the current flow (as in [7]). This notation, claimed by applications to interactive systems, allows us to identify systems which we want to establish a communication with.

Before the description of our logic, in order to better understand the aim in its design, we give the following simplified example. Consider a computer network with some physical links among them and, for simplifying, let us suppose that the possible states of the computers are: ready to send and receive, ready to send but not to receive, not ready to send but ready to receive and, finally, not ready to send and not ready to receive. Assume also that the computers are working changing their states and, in each change of state, a change in the

instant in its time flow is produced. That is, the time flow of each computer represents the different states of this computer with respect to the time course. For simplicity in this example, we reason only with two computers, but these ideas can be easily generalizable for more computers. If a computer X_1 is planning to establish a communication with another X_2 in an instant $t_1 \in T_1$, being T_i the time flow of X_i for $i \in \{1, 2\}$, t_1 has to be a *ready to send* state and X_2 has to be in a *ready to receive* state. However, the specific instant in which the communication is executed is not initially known. As a consequence, the possible communications are represented by a subset of T_2, which is the image of t_1 by the accessibility non deterministic operator. Moreover, if the communication from t_1 is effectively executed in the instant $t_2 \in T_2$, then t_2 is the image of t_1 by the execution function and we assume that the images of every later instant of t_1 are lower bounded by t_2, because in this moment the information in X_1 about X_2 has been updated. The following figures represent two different situations.

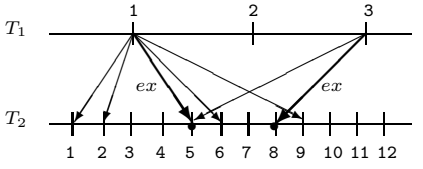


figure 1

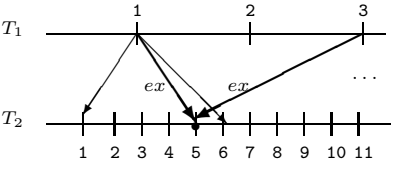


figure 2

In figure 1, the image of instant 1 of T_1 are 1, 2, 5 and 9 in T_2, which are possible instants in ready to receive state. Also, in instant 5 of T_2 a real communication (an execution) occurs. Instant 3 of T_1 is analogous. The image of 3 is lower bounded by the execution instant of 1, that is, 5 of T_2.

In figure 2, two different executions from instants 1 and 3 of T_1 occur in the same instant 5 of T_2. This can be explained because the instant 5 of T_2 can be in a *ready to receive* state for which communication with instant 1 of T_1 occurred, but due to the computer X_2 do not change its state, a communication with 3 of T_1 occurred also.

The figures above give the intuitive idea about the type of frame that we are going to define in this paper: different time flows and the accessibility between every pair of them is given by a lower bounded non-deterministic operator (possible communications) and an execution function (effective communications) which determines these lower bounds. This kind of frame, as we will see, will allow us to interpret temporal×modal connectives of our MAT logic.

This paper is organized as follows: In section 2, concepts of non-deterministic operator and lower bounded non-deterministic operator are introduced. Notation that will be used in the rest of the paper is introduced too. In section 3, MAT logic is defined. Moreover, the semantic is shown, emphasizing the set of *accessibility non deterministic operators* among temporal flows, $\mathcal{C}$, the set of *execution functions*, $\mathcal{F}_{ex}$. In section 4, an axiom system S_{MAT} for our logic is introduced. Also, soundness and completeness of the system are stated. Finally, in section 5, some conclusions and future works are shown.

2 Lower Bounded Unary Non-deterministic Operators

This section is devoted to the necessary preliminaries about non-deterministic operators.

Definition 1. *Let A and B be non-empty sets and $n \in \mathbb{N}$ where $n \geq 1$. Any function $F : A^n \to 2^B$ is said to be a* **non-deterministic operator of arity** *n from A to B. Any non-deterministic operator of arity 1 from A to B is called a* **ndo** *from A to B. The set $\mathcal{N}do(A, B)$ is the set of all non-deterministic operators of arity 1 from A to B.*

In the same way that occurs when we work with functions, $F(X)$ will denote the set $\bigcup_{x \in X} F(x)$, for all $F \in \mathcal{N}do(A, B)$ and $X \subseteq A$.

Definition 2. *Given two non-empty sets A and B, the relation $\subseteq$ in $\mathcal{N}do(A, B)$ is defined by $F \subseteq G$ if and only if $F(a) \subseteq G(a)$ for all $a \in A$.*

Remark 1. Non-determinism condition is about the fact that cardinality of the images is arbitrary, contrarily to functions and deterministic operators. Nevertheless, every (total or partial) function $f : A \to B$ can be identified with an element of $F \in \mathcal{N}do(A, B)$:

$$F : A \to 2^B \qquad \text{and} \qquad F(a) = \begin{cases} \{f(a)\}, & \text{if } f \text{ is defined for } a; \\ \varnothing, & \text{if } f \text{ is not defined for } a. \end{cases}$$

In this work, functions will be considered in this previous way. This fact motivates the following definition.

Definition 3. *Let A and B be non-empty sets and $F \in \mathcal{N}do(A, B)$, we define the* **domain** *of F as the set $Dom(F) = \{a \in A \mid F(a) \neq \varnothing\}$. The* **empty ndo***, denoted by $\emptyset$, is the ndo whose domain is empty, that is, $\emptyset : A \to 2^B$ and $\emptyset(a) = \varnothing$, for all $a \in A$.*

As it was mentioned in the introduction, we are interested in linear temporal flows and particularly in the use of ndos with the characteristics collected in the following definition.

Definition 4. *Let A and B be two linear ordered sets and let F be a ndo from A to B. F is* **lower bounded** *if, for all $a \in Dom(F)$, the minimum of $F(a)$ exists (hereinafter denoted $\min F(a)$). $\mathcal{N}do^{lb}(A, B)$ denotes the set of all lower bounded ndos of arity 1 from A to B, its elements will be called lb-ndo.*

Some notations useful in the rest of the paper are introduced now.

Notation: Let $(A, \leq)$ be a linear ordered set, a be an element of A and $X \subseteq A$.
$[a, \rightarrow) = \{x \in A \mid a \leq x\}$, $X\uparrow = \bigcup_{x \in X} [x, \rightarrow)$ and $X\uparrow^* = \bigcup_{x \in X} (x, \rightarrow)$.
$(a, \rightarrow)$, $(\leftarrow, a]$, $(\leftarrow, a)$, $X\downarrow$ and $X\downarrow^*$ can be analogously defined.

3 The MAT Logic

In this section MAT logic is defined as a family of indexed temporal×modal logics $MAT\text{-}\mathfrak{I} = (L_\mathfrak{I}, \mathcal{M}^\mathfrak{I})$ where $\mathfrak{I}$ is a non-empty numerable set of indexes. The selection of this set determines a specific MAT logic. $L_\mathfrak{I}$ denotes the language and $\mathcal{M}^\mathfrak{I}$ the set of models for $L_\mathfrak{I}$.

3.1 The Language $L_\mathfrak{I}$ of $MAT\text{-}\mathfrak{I}$

Given a denumerable set of indexes $\mathfrak{I}$, the alphabet of $L_\mathfrak{I}$ consists of:

(i) a denumerable set, $\mathcal{V}$, of propositional variables;
(ii) the logic constants $\top$ ("truth") and $\bot$ ("falseness"), and the boolean connectives $\neg$ ("not"), $\wedge$ ("and"), $\vee$ ("or") and $\rightarrow$ ("if... then..."");
(iii) the temporal connective of future G ("it will always be that") and H ("it has been always that");
(iv) the three indexed modal connectives $<i>$, $<i>_{min}$ and $<i>_{ex}$ for $i \in \mathfrak{I}$;
(vi) the auxiliary symbols: (,).

The well formed formulae (*wffs*) are generated by the construction rules of classical propositional logic by adding the new rule:

If A is a wff, *then GA, HA. $<i>A$, $<i>_{min}A$ and $<i>_{ex}A$ are* wffs. The desired interpretation of the new modal connectives is as follows:

- $<i>A$ is read as "*There exists a temporal flow T_i and there exist some states in T_i that are available from present state and A is true in some of these states*".
- $<i>_{min}A$ is read as "*There exists a temporal flow T_i and there exist some states in T_i that are available from present state and A is true in the minimum of these states*".
- $<i>_{ex}A$ is read as "*There exists a temporal flow T_i and there exist some states in T_i that are available from present state and A is true in one of these states, specifically in the execution state*".

We also consider the connectives $[i]$, $[i]_{min}$ and $[i]_{ex}$ as usual in modal logic.

3.2 Semantics of $MAT\text{-}\mathfrak{I}$

As we have said in the introduction section, the frames must satisfy some properties formalized in the following definition.

Definition 5. *A MAT-* **frame** *is a tuple $\Sigma = (W, \Lambda, \mathcal{T}, \mathcal{C}, \mathcal{F}_{ex})$ such that:*

(1) W is a non-empty set (set of labels that will be used for temporal flows).
(2) Λ is a distinguished subset (possibly empty) of W.
(3) $\mathcal{T} = \{(T_w, <_w) \mid w \in W\}$ is a non-empty set of temporal flows, such that $T_w \neq \varnothing$ for all $w \in W$; $T_w \cap T_{w'} = \varnothing$ for all $w, w' \in W$ with $w \neq w'$ and, for all $w \in W$, $<_w$ is a strict order relation in T_w which is linear. The elements t_w of the disjoint union $\mathcal{C}oord_\Sigma = \bigoplus_{w \in W} T_w$ are called **coordinates**.

(4) $\mathcal{C}$ *is a set of ndos* $\mathcal{C} = \{C_w^l \mid (w,l) \in W \times \Lambda\}$ *whose elements, called* **accessibility ndos**, *satisfy that for any* $(w,l) \in W \times \Lambda$, C_w^l *is an lb-ndo (possibly the empty ndo,* $\emptyset$*) from* T_w *to* T_l.

(5) $\mathcal{F}_{ex}$ *is a set of partial functions* $\mathcal{F}_{ex} = \{\xrightarrow{w\ l}_{ex} \mid (w,l) \in W \times \Lambda\}$, *whose elements, called* **execution functions**, *satisfy that for any* $(w,l) \in W \times \Lambda$, $\xrightarrow{w\ l}_{ex}$ *is a partial function (possibly the empty function,* $\emptyset$*) from* T_w *to* T_l.

(6) $\mathcal{C}$ *and* $\mathcal{F}_{ex}$ *satisfy the following conditions:*

6.1) $\xrightarrow{w\ l}_{ex} \subseteq C_w^l$ *for all* $(w,l) \in W \times \Lambda$

6.2) If $t_w \in Dom(\xrightarrow{w\ l}_{ex})$, *then* $C_w^l\Big((t_w, \rightarrow)\Big) \subseteq \big(\xrightarrow{w\ l}_{ex}(t_w)\big)\uparrow$

Remark 2. Condition (6) gives the relation between the accessibility ndos and execution functions: (6.1) says that the image of the execution function is a subset of the image of the corresponding ndo, that is, the execution instant in one of the available states and (6.2) represents that the execution instant is a lower bound for the image by the ndo of later states, as we have said previously.

The following definition introduces a lb-ndo useful in the rest of the paper.

Definition 6. *Given a MAT-frame* $(W, \Lambda, \mathcal{T}, \mathcal{C}, \mathcal{F}_{ex})$ *and* $(w,l) \in W \times \Lambda$, *we define the lb-ndo,* $m_l : T_w \to 2^{T_l}$, *as follows:*

$$m_l(t_w) = \begin{cases} \{\min\ C_w^l(t_w)\}, & \text{if } t_w \in Dom(C_w^l) \\ \varnothing, & \text{otherwise} \end{cases}$$

Now, we have all the necessary elements to define semantics of the connectives of $L_{\mathfrak{I}}$. $\mathfrak{I}$ will play a notable role in this semantic (as an arsenal of names for denoting the range of lb-ndos).

Definition 7. *A* **model** *for* $L_{\mathfrak{I}}$ *is an ordered pair* $\mathcal{M}^{\mathfrak{I}} = (\Sigma^{\mathfrak{I}}, h)$, *where:*

i) $\Sigma^{\mathfrak{I}}$ *is a MAT-frame,* $\Sigma^{\mathfrak{I}} = (W, \Lambda^{\mathfrak{I}}, \mathcal{T}, \mathcal{C}, \mathcal{F}_{ex})$, *such that* $\Lambda^{\mathfrak{I}} = W \cap \mathfrak{I}$. *From now on,* $\Sigma^{\mathfrak{I}}$ *will be called a* **MAT-frame depending on** $\mathfrak{I}$.

ii) A function $h : \mathcal{V} \longrightarrow 2^{\mathcal{C}oord_{\Sigma^{\mathfrak{I}}}}$, *assigning each atom* $p \in \mathcal{V}$ *a subset of* $\mathcal{C}oord_{\Sigma^{\mathfrak{I}}}$ *is called an* **interpretation**.

The interpretation h is recursively extended to a function (still denoted h) defined on all formulae of $L_{\mathfrak{I}}$ *that satisfies usual conditions for boolean and temporal connectives. Moreover, for our special modal connectives, we have:*

$$h(<i>A) = \begin{cases} \{t_w \in \mathcal{C}oord_{\Sigma^{\mathfrak{I}}} \mid C_w^i(t_w) \cap h(A) \neq \varnothing\} & \text{if } i \in \Lambda^{\mathfrak{I}}, \\ \varnothing & \text{otherwise;} \end{cases}$$

$$h(<i>_{min} A) = \begin{cases} \{t_w \in \mathcal{C}oord_{\Sigma^{\mathfrak{I}}} \mid m_i(t_w) \cap h(A) \neq \varnothing\} & \text{if } i \in \Lambda^{\mathfrak{I}}, \\ \varnothing & \text{otherwise;} \end{cases}$$

$$h(<i>_{ex} A) = \begin{cases} \{t_w \in \mathcal{C}oord_{\Sigma^{\mathfrak{I}}} \mid \xrightarrow{w\ i}_{ex}(t_w) \cap h(A) \neq \varnothing\} & \text{if } i \in \Lambda^{\mathfrak{I}}, \\ \varnothing & \text{otherwise;} \end{cases}$$

Remark 3. As expressed in Remark 1, $\xrightarrow{w\ i}_{ex}(t_w)$ is considered as a subset of T_i.

Now, we have the necessary elements for the formal definition of the concepts validity, truth and satisfiability.

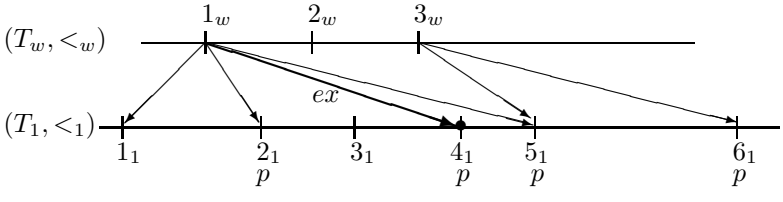
Definition 8. *We say that a formula A of $L_\mathfrak{I}$ is* **satisfiable** *if there exists a model $\mathcal{M}^\mathfrak{I} = (\Sigma^\mathfrak{I}, h)$ for $L_\mathfrak{I}$, and $t_w \in Coord_{\Sigma^\mathfrak{I}}$ such that $t_w \in h(A)$; in this case we also say that A is* **true at** *t_w. A is said to be* **valid in a model** *$\mathcal{M}^\mathfrak{I} = (\Sigma^\mathfrak{I}, h)$ for $L_\mathfrak{I}$, if A is true at every coordinate in the model, that is, if $h(A) = Coord_{\Sigma^\mathfrak{I}}$. Let $\Sigma^\mathfrak{I}$ be a MAT-frame depending on $\mathfrak{I}$. We say that A is* **valid in** *$\Sigma^\mathfrak{I}$, if A is valid in every model $\mathcal{M}^\mathfrak{I} = (\Sigma^\mathfrak{I}, h)$. Let $\mathbb{K}$ be a class of MAT-frames depending on $\mathfrak{I}$. We say that A is* **valid in the class** *$\mathbb{K}$ if A is valid in all MAT-frames $\Sigma^\mathfrak{I} \in \mathbb{K}$. If $\mathbb{K}$ is the class of all MAT-frames depending on $\mathfrak{I}$, then we simply say that A is* **valid**.

The following example allows us to comment the semantics of the specific modal connectives. Given a language $L_\mathfrak{I}$ being $\mathfrak{I} = \{0, 1\}$, we define the MAT-frame depending on $\mathfrak{I}$, $\Sigma^\mathfrak{I} = (W, \Lambda^\mathfrak{I}, \mathcal{T}, \mathcal{C}, \mathcal{F}_{ex})$ such that:

- $W = \{w, 1\}$, thus by definition we have that $\Lambda^\mathfrak{I} = \{1\}$.
- $\mathcal{T} = \{(T_w, <_w), (T_1, <_1)\}$, where $T_w = \{1_w, 2_w, 3_w\}$, $T_1 = \{1_1, 2_1, 3_1, 4_1, 5_1, 6_1\}$ and $<_w, <_1$ the usual strict linear order relations in T_w and T_1, respectively.
- $\mathcal{C} = \{C_1^1, C_w^1\}$, being $C_1^1 = \emptyset_1^1$, $C_w^1(1_w) = \{1_1, 2_1, 4_1, 5_1\}$, $C_w^1(2_w) = \varnothing$ and $C_w^1(3_w) = \{5_1, 6_1\}$
- $\mathcal{F}_{ex} = \{\xrightarrow{1\ 1}_{ex}, \xrightarrow{w\ 1}_{ex}\}$, where $\xrightarrow{1\ 1}_{ex} = \emptyset_1^1$ and $\xrightarrow{w\ 1}_{ex}$ is defined by: $\xrightarrow{w\ 1}_{ex}(1_w) = \{4_1\}$; $\xrightarrow{w\ 1}_{ex}(2_w) = \varnothing$; $\xrightarrow{w\ 1}_{ex}(3_w) = \varnothing$.

Consequently we have $m_1(1_w) = \{1_1\}$; $m_1(2_w) = \varnothing$; $m_1(3_w) = \{5_1\}$ and, for all $t_1 \in T_1$, $m_1(t_1) = \varnothing$.

If p is a formula, we can define a model $(\Sigma^\mathfrak{I}, h)$ such that $h(p) = \{2_1, 4_1, 5_1, 6_1\}$. The following figure represents this model.



In this model, statements about truth or falseness of some formulae in coordinates of T_w are collected in the following table:

$<1> p$ is true in 1_w and is true in 3_w	$<1> p$ is false in 2_w
$<1>_{min} p$ is true in 3_w	$<1>_{min} p$ is false in 1_w and 2_w
$<1>_{ex} p$ is true in 1_w	$<1>_{ex} p$ is false in 2_w and 3_w
$[1] p$ is true in 2_w, 3_w and in all $t_1 \in T_1$	$[1] p$ is false in 1_w
$[1]_{min} p$ is true in 2_w, 3_w and in all $t_1 \in T_1$	$[1]_{min} p$ is false in 1_w
$[1]_{ex} p$ is true in all $t_w \in T_w$ and in all $t_1 \in T_1$	

In the context of computer network, if we interpret the formula p by *the computer is running a specific program* P then, for example, in 1_w the formula $F[\,1\,]\,p$ is true and $H\ <1>_{ex}\ p$ is true in 2_w. The first formula means that in a future state (in this case 3_w), all the available states in T_1 (5_1 and 6_1) will be running the program P. The second formula means that always in the past of 2_w (that is, in 1_w) the execution state in T_1 (in this case, 4_1) was running the program P.

4 The Axiom System $\mathcal{S}_{MAT}$

In this section an axiom system for the language $L_{\mathfrak{I}}$ is introduced denoted by $\mathcal{S}_{MAT}$. This system has the following schema of axioms and inference rules.

1. Propositional linear temporal logic schema $\mathcal{K}_l$
2. For each $i \in \mathfrak{I}$ standard modal schemas of axioms:
   2.1 $[\,i\,](A \to B) \to ([\,i\,]A \to [\,i\,]B)$.
   2.2 $[\,i\,]_{min}(A \to B) \to ([\,i\,]_{min}A \to [\,i\,]_{min}B)$.
   2.3 $[\,i\,]_{ex}(A \to B) \to ([\,i\,]_{ex}A \to [\,i\,]_{ex}B)$.
3. For each $i \in \mathfrak{I}$, specific schemas of axioms:
   3.1 $<i>\,A \to <i>_{\,min}(A \vee FA)$
   3.2 $<i>_{\,min}A \to [\,i\,]_{min}A$
   3.3 $<i>_{\,ex}(A \wedge GA) \to G[\,i\,]A$
   3.4 $<i>_{\,ex}A \to [\,i\,]_{ex}A$
   3.5 $(\lambda\, A \wedge \lambda'\, B) \to \lambda\,(A \wedge (PB \vee B \vee FB))$, where:

$$\begin{cases} (\dagger)_1 & \lambda = \gamma_1 <j_1>_{\,min}\gamma_2 \ldots <j_n>_{\,min} \quad \text{with} \begin{cases} \gamma_l \in \{P, F, \epsilon\}, \\ j_l \in \mathfrak{I}, \text{ and} \\ 1 \le l \le n. \end{cases} \\ (\dagger)_2 & \lambda' = \gamma'_1 <k_1>_{\,min}\gamma'_2 \ldots <k_s>_{\,min} \quad \text{with} \begin{cases} \gamma'_l \in \{P, F, \epsilon\}, \\ k_l \in \mathfrak{I}, \text{ and} \\ 1 \le l \le s. \end{cases} \\ (\dagger)_3 & j_n = k_s \end{cases}$$

   3.6 $<i>_{\,min}A \to <i>\,A$
   3.7 $<i>_{\,ex}A \to <i>\,A$

The inference rules are propositional linear temporal logic $\mathcal{K}_l$ inference rules together with the rule: For all $i \in \mathfrak{I}$: $\dfrac{A}{[\,i\,]A}$

Informal reading of specific modal axioms is the following:

3.1 and 3.2 formalize existence and uniqueness of the minimum available state, respectively.
3.3 ensures that available states from a given one are lower bounded by execution states.
3.4 means that if there exists the execution instant then it is unique.

3.5 say that every two access chains to the same time flow (because $j_n = k_s$) through the minima converge.
3.6 and 3.7 ensure that minimum and execution states are available states, respectively.

Syntactical concepts as *Proof* or *Theorem* are defined as usual.

Theorem 1. *System S_{MAT} is sound and complete.*

The soundness of S_{MAT} can be obtained by proving the validity of the axioms and taking into account that the inference rules are validity-preserving.

Regarding completeness, a *step-by-step* proof (see, for example, [2,4,6,7]) can be given in the following terms: Given any consistent formula A , we have to prove that A is satisfiable. With this purpose, the step-by-step method defines a MAT-frame Σ and a function Φ_Σ which assigns maximal consistent sets to any coordinate, such that $A \in \Phi_\Sigma(t_w)$ for some $t_w \in \mathcal{C}oord_\Sigma$. The process to build such a frame is recursive, successive extensions of the frames are defined until Σ is obtained.

Due to lack of space, the formal details of soundness and completeness proofs are left for a longer version of this paper.

5 Conclusions and Future Work

A new combination of modal and temporal logic, called MAT logic has been presented. This logic allows us to manage communications among systems without synchronization restrictions. The achievement of this goal has been possible due to the use of non-deterministic operators among time flows. Together with the semantic of the MAT logic, defined in an algebraic style, a sound and complete axiom system S_{MAT} has been introduced.

At the present time, we are looking for formulae whose validity characterizes important properties for the communications of systems to extend the field of application of our MAT logic, for example in the planning area.

We are also studying the possibility of increasing the expressivity of MAT-Logic combining a totally expressive temporal logic (concretely, LN Logic [5]) with a modal logic in the same way used in this paper.

Last bust not least, it is planned to design a method for automated deduction in MAT-Logic.

References

1. Davide Ancona and Viviana Mascardi. Coo-bdi: Extending the bdi model with cooperativity. In João Alexandre Leite, Andrea Omicini, Leon Sterling, and Paolo Torroni, editors, *DALT*, volume 2990 of *Lecture Notes in Computer Science*, pages 109–134. Springer, 2003.
2. Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 2001.

3. Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zhisheng Huang, and Leendert van der Torre. The boid architecture: conflicts between beliefs, obligations, intentions and desires. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 9–16, 2001.
4. John P. Burgess. Basic tense logic. In *Handbook of philosophical logic, Vol. II*, volume 165 of *Synthese Lib.*, pages 89–133. Reidel, Dordrecht, 1984.
5. Alfredo Burrieza and Inmaculada P. de Guzmán. A new algebraic semantic approach and some adequate connectives for computation with temporal logic over discrete time. *J. Appl. Non-Classical Logics*, 2(2):181–200, 1992.
6. Alfredo Burrieza and Inmaculada P. de Guzmán. A functional approach for temporal modal logics. *Acta Inf.*, 39(2):71–96, 2003.
7. Alfredo Burrieza, Inmaculada P. de Guzmán, and Emilio Muñoz. Indexed flows in temporal×modal logic with functional semantics. In *Proceedings Ninth International Symposium on Temporal Representation and Reasoning (TIME'02)* , pages 146–153, 2002.
8. Rocco De Nicola and Michele Loreti. A modal logic for mobile agents. *ACM Trans. Comput. Logic*, 5(1):79–128, 2004.
9. Hans-Dieter Ehrich, Carlos Caleiro, Amílcar Sernadas, and Grit Denker. Logics for specifying concurrent information systems. In Jan Chomicki and Gunter Saake, editors, *Logics for Databases and Information Systems*, pages 167–198. Kluwer, 1998.
10. A. J. Dix. *Formal Methods in Human-Computer interaction* . Chapter 4: Nondeterminism as a paradigm for understanding the user interface, pages 97-127. Cambridge University Press, 1990.
11. Joseph Y. Halpern and Moshe Y. Vardi. *Reasoning about Knowledge and Time in Asynchronous Systems*. ACM: Association for Computer Machinery. pp 53-65. 1988.
12. Chris W. Johnson. Using temporal logic to prototype interactive systems. In Dan Diaper, David J. Gilmore, Gilbert Cockton, and Brian Shackel, editors, *INTERACT*, pages 1019–1020. North-Holland, 1990.
13. Zohar Manna and Amir Pnueli. *The temporal logic of reactive and concurrent systems*. Springer-Verlag, New York, 1992. Specification.
14. Andrea Masini and Andrea Maggiolo-Schettini. Ttl: A formalism to describe local and global properties of distributed systems. *Informatique Theorique et Applications*, 26:115–162, 1992.
15. Alan Mishchenko and Robert K. Brayton. A theory of non-deterministic networks. In *ICCAD*, pages 709–717. IEEE Computer Society / ACM, 2003.
16. R. Ramanujam. Locally linear time temporal logic. In *LICS*, pages 118–127, 1996.
17. Anand S. Rao and Michael P. Georgeff. An abstract architecture for rational agents. In *KR*, pages 439–449, 1992.
18. Pazhamaneri S. Thiagarajan and Jesper G. Henriksen. Distributed versions of linear time temporal logic: A trace perspective. *Lecture Notes in Computer Science*, 1492:643–681, 1998.
19. Richmond H. Thomason. Combinations of tense and modality. In *Handbook of philosophical logic, Vol. II*, volume 165 of *Synthese Lib.*, pages 135–165. Reidel, Dordrecht, 1984.
20. Masaru Tomita. *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1985.
21. Paolo Torroni. Computational logic in multi-agent systems: Recent advances and future directions. *Annals of Mathematics and Artificial Intelligence*, 42(1-3): 293–305, 2004.

Probabilistic Logic with Strong Independence

Fabio G. Cozman[1], Cassio P. de Campos[1,2], and José Carlos F. da Rocha[3]

[1] Escola Politécnica, Univ. de São Paulo, São Paulo, SP, Brazil
[2] Univ. Mackenzie, São Paulo, SP, Brazil
[3] Univ. Estadual de Ponta Grossa, Ponta Grossa, PR, Brazil
fgcozman@usp.br, cassio@ime.usp.br, jrocha@uepg.br

Abstract. This papers investigates the manipulation of statements of strong independence in probabilistic logic. Inference methods based on polynomial programming are presented for strong independence, both for unconditional and conditional cases. We also consider graph-theoretic representations, where each node in a graph is associated with a Boolean variable and edges carry a Markov condition. The resulting model generalizes Bayesian networks, allowing probabilistic assessments and logical constraints to be mixed.

1 Introduction

Probabilistic logic offers a unifying language for a vast portion of human and computational discourse, as it merges logical and probabilistic sentences into a uniform scheme. However, probabilistic logic faces two difficulties: *inferential vacuousness* and *computational complexity*.

A simple example of inferential vacuousness is as follows. Suppose events A and B have no logical relation and $P(A) = p$, $P(B) = q$, with $p, q > 0$. Then the probability of $P(A \wedge B)$ is *completely* vacuous: $P(A \wedge B)$ can be in the whole interval $[0, 1]$. In this simple example it is obvious that a judgement of independence would greatly change matters: if the events A and B are considered independent, then $P(A \wedge B) = pq$. In fact, judgements of independence are used in large multivariate models, such as Bayesian networks, to create complex distributions out of few probability assessments.

The goal of this paper is to pursue a language that retains both the freedom of probabilistic logic and the power of independence relations. In short, the thesis explored in this paper is that independence relations should be part of the vocabulary of general probabilistic logic and probabilistic logic programming. The adopted language is the propositional part of Halpern's first order probabilistic logic [1,2], plus a few predicates that are needed to deal with independence. The contribution is to propose concrete algorithms that can handle this extended vocabulary.

As for computational complexity, probabilistic logic without independence relations is already quite hard. Independence relations cannot per si reduce this complexity; however, independence relations may allow us to build compact and modular knowledge bases, hopefully suggesting simplifications and approximations that are not available in independence-free probabilistic logic. This paper explores graph-theoretic representations for large multivariate models, aiming at a language that can expose such modularity in practice.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 612–621, 2006.

Section 2 presents a few relevant facts about probabilistic logic. Section 3 treats the inference problem with strong independence judgements and unstructured sentences, while Section 4 defines a graph-theoretic representation for probabilistic and logical assessments and presents an inference algorithm. Section 5 concludes the paper.

2 Probabilistic Logic

In this paragraph we fix basic notation; there are detailed treatments of propositional and first order logic in the literature [3]. We use a set of *propositions* $\mathbb{A} = \{A_i\}_{i=1}^n$, taken as facts, situations, or events. An *atomic formula* consists of a proposition. A *formula* consists either of an atomic formula or of a combination of other formulas through logical connectives: negation ($\neg$), disjunction ($\vee$) and conjunction ($\wedge$). Connectives have the usual semantics defined by truth tables. A *literal* is a formula consisting either of a proposition or a negated proposition. A *truth assignment* to a set of propositions is a vector assigning either value true or value false to each proposition (and each proposition can take only one of these values). These assignments are often called *possible worlds*. For n propositions, there are 2^n truth assignments; if a formula ϕ is true for some truth assignment w, then ϕ is satisfied by w.

The *probabilistic satisfiability* problem is this: given a set of propositions $\mathbb{A}$, a set of formulas $\{\phi_i\}_{i=1}^m$ over $\mathbb{A}$, and a set of assessments p_i in the real interval $[0, 1]$, such that each p_i is associated with a formula ϕ_i, is there a probability distribution P such that $P(\phi_i) = p_i$ for all p_i [4,5]? Here $P(\phi_i)$ is taken as the measure of the set of possible worlds where ϕ_i is true. A related problem is to determine the probability $P(\phi^*)$ for a formula ϕ^* given a set of formulas and assessments. This is often called a *probabilistic inference* for $P(\phi^*)$.

Probabilistic satisfiability has old roots and has been rediscovered a few times. Boole tried already to combine probability and logic [6]; Boole's efforts were then generalized by Hailperin [7], and later proposed in the artificial intelligence literature by Nilsson [8]. Linear programming is the main tool for satisfiability and inference, and it can handle assessments of conditional probability, inference of conditional probability, and assessments in the form of probability intervals. An excellent historical and technical review is given by Hansen and Jaumard [5].

A different trail to probabilistic satisfiability started with de Finetti, who began a comprehensive program to place logical and probabilistic statements into a single coherency-based framework [9]. Linear programming is again the main inference tool [10], but the coherency-based approach is distinguished by its ability to handle conditioning on propositions of zero probability [11] — in this paper we assume that every conditioning event has positive probability. The relationship between probabilistic satisfiability and the coherency-based framework has been explored recently [12].

A few authors have considered *stochastic independence*[1] in probabilistic satisfiability [13,5]. No systematic implementation of proposed algorithms has been discussed in the literature. The study of independence in the context of coherency-based reasoning, where zero probabilities are dealt with explicitly, started recently with the seminal work

[1] Events C and D are stochastically independent if $P(C|D) = P(C)$ and $P(D|C) = P(D)$.

of Coletti and Scozzafava [14], and is explored in depth in the work of Vantaggi [15]. Specialized algorithms for small problems have been proposed in this latter work.

Two further notable generalizations of probabilistic satisfiability have appeared in the artificial intelligence literature: constructs from first order logic have been incorporated [1], and declarative programs have adopted techniques from probabilistic satisfiability [16,17]. A rather elegant formulation of *probabilistic logic programming* is given by Lukasiewicz, where a rule is expressed as $(\phi|\varphi)[l,u]$, meaning "the probability of ϕ conditional on φ is in the interval $[l,u]$" [18]. We adopt this notation in this paper, specialized for propositional formulae. More general first order probabilistic logics have appeared in the work of Bacchus [19] and Halpern [1,2]. These logics define model-based semantics for general formulae; in this paper we only use their propositional parts plus predicates that indicate independence (say an independent predicate) and that indicate graph-theoretic constructs (such as parentOf).

3 Inferences with Unstructured Sentences

Consider m assessments $(\phi_i|\varphi_i)[l_i,u_i]$ defined on n Boolean variables $X_1,\ldots,X_n$. We use x to indicate that variable X is true, and use $\overline{x}$ to indicate that variable X is false. There are 2^n possible worlds (each formula ϕ_i, φ_i is true or false in each world) and world w_k is associated with probability p_k. Denote by $V[\phi]$ the vector with 2^n elements where the kth element is 1 if ϕ is true in w_k and 0 otherwise, and denote by $\mathbf{p}$ the vector containing all p_k. Under the assumption of positivity for conditioning events mentioned previously, any conditional assessment can be written as constraints such as $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i]) \cdot \mathbf{p} \geq 0$.

The inference problem is to compute the lower and upper probabilities for a formula ϕ^* conditional on a formula φ^*:

$$\min_{\mathbf{p}} / \max_{\mathbf{p}} \left(V[\phi^* \wedge \varphi^*] \cdot \mathbf{p}\right) / \left(V[\varphi^*] \cdot \mathbf{p}\right), \tag{1}$$

subject to $\sum_k p_k = 1$, $p_k \geq 0$ for all k, $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i]) \cdot \mathbf{p} \geq 0$ and $(V[\phi_i \wedge \varphi_i] - u_i V[\varphi_i]) \cdot \mathbf{p} \leq 0$, for all $i = 1,\ldots,m$. The Charnes-Cooper transformation can reduce this optimization problem to the linear program $\min_{\mathbf{p}} / \max_{\mathbf{p}} \left(V[\phi^* \wedge \varphi^*] \cdot \mathbf{p}\right)$, subject to $V[\varphi^*] \cdot \mathbf{p} = 1$, $p_k \geq 0$ for all k, $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i]) \cdot \mathbf{p} \geq 0$ and $(V[\phi_i \wedge \varphi_i] - u_i V[\varphi_i]) \cdot \mathbf{p} \leq 0$, for all $i = 1,\ldots,m$. The Charnes-Cooper transformation is well-known in probabilistic logic [5] and statistics [20].

Note that probabilistic logic is essentially concerned with sets of probability measures — sets induced by linear and fractional equality and inequality constraints. To this model we will include independence judgements. There are several concepts of independence that can be used when one deals with sets of probabilities [21,22,23]. We adopt the most commonly used concept of *strong independence*: formulas θ and ϑ are *strongly independent* conditional on η when

$$\left(V[\theta \wedge \vartheta \wedge \eta] \cdot \mathbf{p}\right) \left(V[\eta] \cdot \mathbf{p}\right) = \left(V[\theta \wedge \eta] \cdot \mathbf{p}\right) \left(V[\vartheta \wedge \eta] \cdot \mathbf{p}\right). \tag{2}$$

Suppose now that s statements of the form $SIN(\theta_j, \vartheta_j | \eta_j)$ are present in the knowledge base, indicating that θ_j and ϑ_j are strongly independent given η_j. Each one of

these independent statements implies a non-linear expression (2), given the positivity assumption and the fact that multilinear programs have global optima in the boundary of the feasible region [24]. Thus *s polynomial* equalities of the form (2) must be added to problem (1). We can avoid the fractional term in the objective function without changing the basic properties of the problem:

Theorem 1. *Problem (1) with additional constraints (2) is equivalent to*

$$\min_{\mathbf{p}'} / \max_{\mathbf{p}'} \left(V[\phi^* \wedge \varphi^*] \cdot \mathbf{p}'\right), \tag{3}$$

subject to constraints (2), $V[\varphi^*] \cdot \mathbf{p}' = 1$*,* $p_k \geq 0$ *for all* k*,* $(V[\phi_i \wedge \varphi_i] - l_i V[\varphi_i]) \cdot \mathbf{p}' \geq 0$ *and* $(V[\phi_i \wedge \varphi_i] - u_i V[\varphi_i]) \cdot \mathbf{p}' \leq 0$*, for all* $i = 1, \ldots, m$*.*

Proof. Introduce a new variable t such that $t^{-1} = (V[\varphi^*] \cdot \mathbf{p})$. Consider a new vector $\mathbf{p}' = \mathbf{p}t$. The objective function becomes $(V[\phi^* \wedge \varphi^*] \cdot (\mathbf{p}'/t))\, t = (V[\phi^* \wedge \varphi^*] \cdot \mathbf{p}')$. The constraint $t^{-1} = (V[\varphi^*] \cdot \mathbf{p}'/t)$ becomes $V[\varphi^*] \cdot \mathbf{p}' = 1$. As for the strong independence constraints, note that constraints (2) remain the same (but over $\mathbf{p}'$ instead of $\mathbf{p}$). The same applies to the remaining linear constraints. □

Unlike geometric programming problems, constraints (2) lead to nonconvex primal and dual programs. Existing solution methods produce sequences of sub-problems that eventually contain only the global optimum, using either branch-and-bound or cutting-plane techniques [25,26,27,28,29]. The algorithms of Maranas and Floudas [27], and Gochet and Smeers [25] produce convex nonlinear sub-problems, while Sherali and Adams' algorithm produces linear sub-problems [28].

The characteristics of Sherali and Adams' branch-and-bound algorithm make it particularly suitable for probabilistic logic. As the sub-problems generated by this method are linear programs, column generation techniques can be applied to them — and column generation techniques are necessary to handle large scale problems [5]. The idea of Sherali and Adams' algorithm is to replace products of variables by new "artificial" variables, and to solve the resulting linear problem. The algorithm iterates by branching over the range of variables whenever necessary, until each artificial variable is close enough to its corresponding product.

The probabilistic satisfiability problem is NP-Complete [4]. The inclusion of independence statements $SIN(\theta, \vartheta|\eta)$ makes it harder:

Theorem 2. *Problem (1) with additional constraints (2) is* $\mathrm{NP}^{\mathrm{PP}}$*-Hard.*

Proof. We can reduce a binary credal network belief updating problem [30] to this problem, naming each node of the network as a formula ϕ_i, specifying each local probability constraint of the network $P(\phi_i|\mathrm{pa}(\phi_i))$ as a probabilistic logic constraint and inserting (a polynomial number of) independence statements $SIN(\phi_i, \phi_j|\phi_k)$ between any two nodes ϕ_i and ϕ_j that are separated by ϕ_k in the network. As the binary credal network belief updating problem is $\mathrm{NP}^{\mathrm{PP}}$-Complete, this theorem follows. □

4 Graph-Theoretic Representations: PPL Networks

The flexibility of propositional probabilistic logic with independence is attractive but comes at a price in computational complexity. Besides, a language that is too general

and unstructured may overwhelm users with too many possible options. In this section we explore situations where assessments and judgements in probabilistic logic can be compactly organized using graphs — following the practical success of several statistical models based on graphs, such as Bayesian and Markov networks [31].

In Bayesian networks and their many variants, the underlying graphs serve simultaneously as an encoding for independence relations and for assessments. In the context of probabilistic logic it seems reasonable to aim at representations that can accommodate both probabilistic and purely logical assessments. As logical constraints have not "direction," we consider graphs $\mathcal{G}$ where each node is associated with a proposition, and containing directed and undirected edges. The undirected edges are clearly intended to represent logical constraints, as in constraint and mixed networks [32].

Our requirement about the graph $\mathcal{G}$ is that it is a *chain graph*; that is, it does not have any directed cycles [33]. The semantics of the graph is given by the following *Markov condition* over all subsets $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{S}$ of nodes of $\mathcal{G}$ [33, p. 76]: If $\mathbf{S}$ separates $\mathbf{A}$ and $\mathbf{B}$ in the smallest ancestral set containing $\mathbf{A} \cup \mathbf{B} \cup \mathbf{S}$, then $\mathbf{A}$ and $\mathbf{B}$ are independent conditional on $\mathbf{S}$. This condition is equivalent to various other "local" properties when probability values are all nonzero, an assumption we cannot make in the presence of logical constraints. We will always assume that *conditional* probabilities are computed for conditioning events that have positive probability. The assumption of acyclicity and the Markov condition are consistent with the representation of logical constraints by undirected edges. For ease of exposition, we also assume that logical constraints contain only two propositions each.

We call the resulting structures *binary PPL networks*. A similar scheme of assessments has been considered previously by Campos and Cozman [34], where a directed acyclic graph is associated with propositions and arbitrary assessments over logical formulas. Their resulting model is not as convenient as the one developed in this paper; it is a bit restrictive in its reliance on directed acyclic graphs, and it is too liberal in the assessments it accepts, leading to computational difficulties.

There are several algorithms that compute probabilities in standard, probabilistic chain graphs [33]. Algorithms developed for Bayesian networks can be easily adapted to chain graphs, and there is an extensive literature on the former. A particularly simple algorithm is *variable elimination* [35,36]. The purpose of variable elimination is to efficiently compute an expression such as $\sum_{\mathbf{X}} \prod_i P(X_i|\mathrm{pa}(X_i))$, where $\mathbf{X}$ is a set of random variables X_i. In fact variable elimination can be directly applied to inference in chain graphs, where each term in the inner product may be a probability distribution or simply an unnormalized potential. The variable elimination algorithm has been recently employed to conduct inferences in *credal networks*; that is, graph-theoretical structures that are similar to Bayesian networks but where a random variable X_i may be associated to a *set* of probability distributions $K(X_i|\mathrm{pa}(X_i))$ [37,38]. The idea is to first run variable elimination "symbolically" and store the intermediate expressions in the sum/product. These expressions form a multilinear program that is then solved.

The same idea can be applied to our current setting, where one may wish to compute lower and upper conditional probabilities. This result is reached by an algorithm of two stages. In the first stage, the PPL network is transformed in a Boolean credal network that encodes the dependence structure of the original model. In the second stage a

modified version of Campos and Cozman algorithm [37] is used to compute the desired interval.

A few definitions are useful in the remainder of the paper. A *Boolean credal network* is a triple $(\mathcal{G}, \mathbb{X}, \mathbb{K})$ where $\mathcal{G}$ is a directed acyclic graph with each node associated to a Boolean random variable of $\mathbb{X}$ and $\mathbb{K}$ is a collection of credal sets. Arcs represent direct dependencies between variables in $\mathbb{X}$ and nodes are associated with locally specified credal sets $K(X_i|\mathrm{pa}(X_i))$ [39]. We assume that this structure satisfies the following *Markov condition*: every variable is *strongly* independent of its nondescendants nonparents given its parents. Given a credal network, an event of interest $\{X_q = i\}$ and a set of evidences $\mathbb{E}$, the belief updating procedure aims at computing the limits of a interval for $P(X_q = i|\mathbb{E})$. These limits are called the lower and upper probabilities of $\{X_q = i\}$ given $\mathbb{E}$. Currently, there are several algorithms for computing these values exactly [40,37], although the time complexity of the problem has motivated the utilization of approximate algorithms [41,42,43]. Now, a *binary constraint network* [44] is a triple $(\mathcal{H}, \mathbb{X}, \mathbb{C})$ where $\mathcal{H}$ is an undirected graph and $\mathbb{C}$ is denotes a set of binary constraints on pairs of Boolean variables in $\mathbb{X}$. Each binary constraint C in $\mathbb{C}$ is associated with an edge of $\mathcal{H}$. The usual *constraint satisfaction* problem is to determine a instantiation of the variables in $\mathbb{X}$ that is consistent with all constraints [45].

A binary PPL network is a mixture of both credal and constraint networks.

Definition 1. *Let $\mathbb{X}$ be a set of propositional variables and $\mathbb{C}$ a set of binary logical constraints on variables in $\mathbb{X}$. A binary PPL network is a quadruple $(\mathcal{M}, \mathbb{X}, \mathbb{K}, \mathbb{C})$, where $\mathcal{M}$ is a chain graph with arcs encoding conditional probabilities (through local credal sets) and edges encoding binary logical constraints between its variables; that is, each node X_i is associated to a collection of local credal sets $K(X_i|\mathrm{pa}(X_i)) \in \mathbb{K}$ and each undirected edge $E = (X_i, X_j)$ is associated with a binary logical constraint C of $\mathbb{C}$ with probability of being true defined by a credal set.*

The Figure 1 illustrates the structure of a PPL network. Nodes represent propositional variables and arcs denote direct conditional dependency.

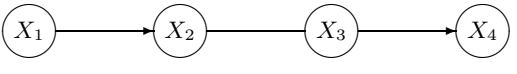


Fig. 1. Example of a binary PPL network

We describe here the two stage procedure for belief updating in binary PPL networks. Suppose that the network of Figure 1 has the following probability distributions associated to it: $P(x_1) = 0.4$; $P(\overline{x_1}) = 0.6$; $P(x_2|x_1) = 0.3$; $P(\overline{x_2}|x_1) = 0.7$; $P(x_2|\overline{x_1}) = 0.5$; $P(\overline{x_2}|\overline{x_1}) = 0.5$; $P(x_4|x_3) = 0.2$; $P(\overline{x_4}|x_3) = 0.8$; $P(x_4|\overline{x_3}) = 0.6$; $P(\overline{x_4}|\overline{x_3}) = 0.4$. Suppose also that it was not possible to produce a Bayesian network because the knowledge engineer could not elicit $P(X_3|X_2)$ or $P(X_2|X_3)$; however suppose it was possible to estimate an interval to $P(X_3)$ represented by the constraint $0.3 \leq P(X_3 = x_3) \leq 0.6$. We will write $P(x_3)$ instead of $P(X_3 = x_3)$ where possible. Additionally, suppose that external data allowed to state that $P(\phi)$ is equal to or greater than 0.252, where $\phi = (x_2 \wedge x_3)$. This constraint is associated to the edge that connects X_2 and X_3 and is not part of a usual credal network.

Now assume we need to compute the maximum possible value of $P(x_4|x_1)$. In the first stage, we transform the binary PPL network into an auxiliary credal network that has the same nodes and directed arcs as the original network, but each undirected edge (X_i, X_j) is replaced by a new artificial node Y, child of X_i and X_j. The local credal sets $K(Y|X_i, X_j)$ are defined through the original credal sets associated to the logical constraint. Figure 2 shows this auxiliary network. C_1 is the artificial node.

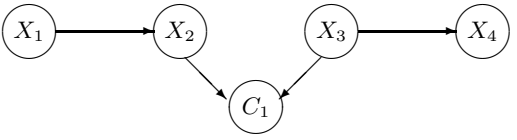


Fig. 2. The auxiliary credal network

In the second stage, a modified version of the multilinear programming algorithm proposed by Campos and Cozman [37] for inference in credal networks is executed.

The multilinear programming problem becomes $\min / \max P(x_4|x_1)$, subject to $P(x_4, x_1) = P(x_4|x_1)(P(x_4, x_1) + P(\overline{x_4}, x_1))$ and

$$
\begin{aligned}
P(X_4, x_1) &= P(c_1, X_4, x_1) + P(\overline{c_1}, X_4, x_1), \text{ where } X_4 \in \{x_4, \overline{x_4}\},\\
P(C_1, X_4, x_1) &= P(C_1, x_1|x_3)P(X_4|x_3)P(x_3) + P(C_1, x_1|\overline{x_3})P(X_4|\overline{x_3})P(\overline{x_3}),\\
&\quad \text{where } C_1 \in \{c_1, \overline{c_1}\}, X_4 \in \{x_4, \overline{x_4}\},\\
P(C_1, X_1|X_3) &= P(C_1|x_2, X_3)P(X_1, x_2) + P(C_1|\overline{x_2}, X_3)P(X_1, \overline{x_2}),\\
&\quad \text{where } C_1 \in \{c_1, \overline{c_1}\}, X_1 \in \{x_1, \overline{x_1}\}, X_3 \in \{x_3, \overline{x_3}\},\\
P(X_1, X_2) &= P(X_2|X_1)P(X_1), \text{ where } X_1 \in \{x_1, \overline{x_1}\}, X_2 \in \{x_2, \overline{x_2}\}
\end{aligned}
$$

and the linear constraints defining the local credal sets. Besides the pure probabilistic assessments, we have $P(c_1) \geq 0.252$ which implies the following additional constraints:

$$
\begin{aligned}
P(c_1) &= P(c_1, x_3) + P(c_1, \overline{x_3}) \geq 0.252,\\
P(c_1, X_3) &= P(c_1, x_1|X_3)P(X_3) + P(c_1, \overline{x_1}|X_3)P(X_3) \text{ where } X_3 \in \{x_3, \overline{x_3}\}.
\end{aligned}
$$

Our implementation promptly produces $P(x_4|x_1) = 0.36$. If the constraint between X_2 and X_3 (the one that implied $P(c_1) \geq 0.252$) is discarded, then we get the interval $\min P(x_4|x_1) = 0.36$ and $\max P(x_4|x_1) = 0.48$. The following result is relevant:

Theorem 3. *The inference in a binary PPL network is NP^{PP}-Complete.*

Sketch of Proof. Hardness comes directly from the fact that a binary PPL network is a extension to the Boolean credal network, so we can trivially reduce the belief updating problem in credal networks to inference in PPL networks. Pertinence is achieved because, when we fix the vertices of the credal sets, we obtain a standard Bayesian network. Then the PP oracle is enough to verify the pertinence. □

5 Conclusion

In this paper we have investigated algorithms for probabilistic logic in the presence of statements of strong independence. We would like to stress the following contributions of the paper. We presented algorithms based on Sherali and Adams' algorithm that produce inferences for large models (compared to the models that can be handled by existing algorithms). We focused on exact inferences, hoping that approximation methods will follow in time. Second, we explored graphical models (in the context of strong independence) that can be used to build large multivariate models in a compact manner. Future work can follow several paths: first order logic constructs, conditioning on zero probability events, coherency-based inference, and general improvement on the efficiency of inference algorithms.

In closing, it is appropriate to discuss the results of this paper in a broader perspective. The strategy here is to combine "classic" probabilistic logic with independence, retaining the freedom usually associated with the former. We do not require that enforced independence relations be specified in any particular fashion; this is to be contrasted with proposals usually labeled "probabilistic relational." In these proposals the idea is to combine logic and probability by restricting the language, so as to obtain Bayesian networks for inference [46,47,48,49,50,51]. While independence-free probabilistic logic may be *too loose*, probabilistic relational models may be *too strict*, as they demand a certain number of independence relations in a specific order. Probabilistic logic with independence seems to be a sensible middle ground; one can either move in the direction of complete generality, or one can build rather specific models. As an example, PPL networks are models that stay between fully general probabilistic logic and probabilistic relational models — accepting that some structure is necessary, but rejecting that a single recipe can be used in every knowledge base.

Acknowledgements

This work has received substantial support from FAPESP through grant 04/09568-0. We thank Barbara Vantaggi and Thomas Lukasiewicz for helping with our understanding of independence in probabilistic logic. The work has also received partial support from CNPq (through grant 3000183/98-4) and HP Brasil R&D.

References

1. Halpern, J.Y.: An analysis of first-order logics of probability. In: International Joint Conference on Artificial Intelligence, Detroit, USA (1989) 1375–1381
2. Halpern, J.Y.: Reasoning about Uncertainty. MIT Press (2003)
3. Nerode, A., Shore, R.A.: Logic for applications. Second edn. Springer Verlag, Ney York (1997)
4. Georgakopoulos, G., Kavvadias, D., Papadimitriou, C.H.: Probabilistic satisfiability. Journal of Complexity **4** (1988) 1–11
5. Hansen, P., Jaumard, B.: Probabilistic satisfiability. Technical report, École Polytechique de Montréal (1996)
6. Boole, G.: The Laws of Thought. Dover edition (1958)

7. Hailperin, T.: Best possible inequalities for the probability of a logical function of events. American Mathematical Monthly **72** (1965) 343–359
8. Nilsson, N.J.: Probabilistic logic. Artificial Intelligence **28** (1986) 71–87
9. de Finetti, B.: Theory of Probability, vol. 1-2. Wiley, New York (1974)
10. Bruno, G., Gilio, A.: Applicazione del metodo del simplesso al teorema fondamentale per le probabilita nella concezione soggettivistica. Statistica **40** (1980) 337–344
11. Coletti, G., Scozzafava, R.: Stochastic independence in a coherent setting. Annals of Mathematics and Artificial Intelligence **35** (2002) 151–176
12. Biazzo, V., Gilio, A., Lukasiewicz, T., Sanfilippo, G.: Probabilistic logic under coherence: Complexity and algorithms. In: International Symposium on Imprecise Probabilities and Their Applications, Shaker (2001) 51–61
13. Andersen, K.A., Hooker, J.N.: Bayesian logic. Decision Support Systems **11** (1994) 191–210
14. Coletti, G., Scozzafava, R.: Zero probabilities in stochastic independence. Information, Uncertainty, Fusion (2000) 185–196
15. Vantaggi, B.: Conditional Independence and Graphical Models in a Coherent Setting. PhD thesis, Università di Perugia, Italy (1999)
16. Lakshmanan, L.V.S., Sadri, F.: Probabilistic deductive databases. In: Symposium on Logic Programming. (1994) 254–268
17. Ng, R., Subrahmanian, V.S.: Probabilistic logic programming. Information and Computation **101(2)** (1992) 150–201
18. Lukasiewicz, T.: Probabilistic logic programming. In: European Conference on Artificial Intelligence. (1998) 388–392
19. Bacchus, F.: Representing and Reasoning with Probabilistic Knowledge. MIT Press, Cambridge, Mass. (1990).
20. Cozman, F.G.: Calculation of posterior bounds given convex sets of prior probability measures and likelihood functions. Journal of Computational and Graphical Statistics **8(4)** (1999) 824–838
21. Couso, I., Moral, S., Walley, P.: A survey of concepts of independence for imprecise probabilities. Risk, Decision and Policy **5** (2000) 165–181
22. de Campos, L., Moral, S.: Independence concepts for convex sets of probabilities. In: Conference on Uncertainty in Artificial Intelligence, San Francisco, Morgan Kaufmann (1995) 108–115
23. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
24. Drenick, R.F.: Multilinear programming: Duality theories. Journal of Optimization Theory and Applications **72(3)** (1992) 459–486
25. Gochet, W., Smeers, Y.: A branch-and-bound method for reversed geometric programming. Operations Research **27(5)** (1979) 983–996
26. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer Verlag, Heidelberg (1996)
27. Maranas, C.D., Floudas, C.A.: Global optimization in generalized geometric programming. Computers and Chemical Engineering **21(4)** (1997) 351–370
28. Sherali, H.D., Adams, W.P.: A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems. Kluwer Academic Publishers, Dordrecht/Boston/London (1999)
29. Tuy, H.: Convex Analysis and Global Optimization. Volume 22 of Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht (1998)
30. de Campos, C.P., Cozman, F.G.: The inferential complexity of Bayesian and credal networks. In: International Joint Conference on Artificial Intelligence. (2005) 1313–1318
31. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, California (1988)

32. Dechter, R., Mateescu, R.: Mixtures of deterministic-probabilistic networks and their AND/OR search space. In: Conference on Uncertainty in Artificial Intelligence, Arlington, Virginia, AUAI Press (2004) 120–129
33. Cowell, R., Dawid, P., Lauritzen, S., Spiegelhalter, D.: Probabilistic Networks and Experts Systems. Springer-Verlag, New York (1999)
34. de Campos, C.P., Cozman, F.G.: Belief updating and learning in semi-qualitative probabilistic networks. In: Conference on Uncertainty in Artificial Intelligence. (2005) 153–160
35. Zhang, N.L., Poole, D.: A simple approach to bayesian network computations. In: Canadian Conference on Artificial Intelligence. (1994) 171–178
36. Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. In Horvitz, E., Jensen, F., eds.: Conference on Uncertainty in Artificial Intelligence, San Francisco, Morgan Kaufmann Publishers (1996) 211–219
37. de Campos, C.P., Cozman, F.G.: Inference in credal networks using multilinear programming. In: Second Starting AI Researcher Symposium, Valencia, IOS Press (2004) 50–61
38. Cozman, F.G., de Campos, C.P., Ide, J.S., da Rocha, J.C.F.: Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments. In: Conference on Uncertainty in Artificial Intelligence, Banff, AUAI Press (2004) 104–111
39. Cozman, F.G.: Credal networks. Artificial Intelligence **120** (2000) 199–233
40. da Rocha, J.C.F., Cozman, F.G.: Inference in credal networks with branch-and-bound algorithms. In: International Symposium on Imprecise Probability and Their Applications, Lugano, Switzerland, Carleton Scientific (2003)
41. da Rocha, J.C.F., Cozman, F.G., de Campos, C.P.: Inference in polytrees with sets of probabilities. In: Conference on Uncertainty in Artificial Intelligence, New York, Morgan Kaufmann (2003) 217–224
42. Ide, J.S., Cozman, F.G.: Approximate inference in credal networks by variational mean field methods. In: International Symposium on Imprecise Probability and Their Applications. (2005) 203–212
43. Cozman, F.G., de Campos, C.P.: Local computation in credal networks. In: Workshop on Local Computation for Logics and Uncertainty, Valencia, IOS Press (2004) 5–11
44. Ladkin, P.B., Maddux, R.: On binary constraint problems. Journal of the ACM **41(3)** (1994) 435–469
45. van Beek, P., Dechter, R.: On the minimality and global consistency of row-convex constraint networks. Journal of the ACM **42(3)** (1995) 543–561
46. Gettor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of relational structure. In: International Conference on Machine Learning. (2001) 170–177
47. Glesner, S., Koller, D.: Constructing flexible dynamic belief networks from first-order probabilistic knowledge bases. Symbolic and Quantitative Approaches to Reasoning with Uncertainty (1995) 217–226
48. Haddawy, P., Suwandi, M.: Decision-theoretic refinement planning using inheritance abstraction. In: AAAI Spring Symposium Series: Decision-Theoretic Planning. (1994)
49. Jaeger, M.: Relational Bayesian networks. In: Conference on Uncertainty in Artificial Intelligence, San Francisco, Morgan Kaufmann (1997) 266–273
50. Ngo, L., Haddawy, P.: Answering queries from context-sensitive probabilistic knowledge bases. Theoretical Computer Science **171(1–2)** (1997) 147–177
51. Poole, D.: Probabilistic Horn abduction and Bayesian networks. Artificial Intelligence **64** (1993) 81–129

Bayesian Model Combination and Its Application to Cervical Cancer Detection

Miriam Martínez[1], Luis Enrique Sucar[2], Hector Gabriel Acosta[3], and Nicandro Cruz[3]

[1] Tecnológico de Acapulco, Acapulco, Guerrero, México
miriamma_ds@hotmail.com
[2] INAOE, Tonantzintla, Puebla, México
esucar@inaoep.mx
[3] Universidad Veracruzana, Xalapa, Veracruz, México
{heacosta, ncruz}@uv.mx

Abstract. We have developed a novel methodology to combine several models using a Bayesian approach. The method selects the most relevant attributes from several models, and produces a Bayesian classifier which has a higher classification rate than any of them, and at the same time is very efficient. Based on conditional information measures, the method eliminates irrelevant variables, and joins or eliminates dependent variables; until an optimal Bayesian classifier is obtained. We have applied this method for diagnosis of precursor lesions of cervical cancer. The temporal evolution of the color changes in a sequence of colposcopy images is analyzed, and the resulting curve is fit to an approximate model. In previous work we develop 3 different mathematical models to describe the temporal evolution of each image region, and based on each model to detect regions that could have cancer. In this paper we combine the three models using our methodology and show very high accurracy for cancer detection, superior to any of the 3 original models.

1 Introduction

In this work we propose a method to combine several models using a Bayesian approach. The method selects the most relevant attributes from several models, and produces a Bayesian classifier that has a high accuracy (improving all the original models), and at the same time is very efficient. The method includes two phases, discretization and structural improvement. Discretization is based on the minimum description length (MDL) principle, so the number of intervals that minimizes the MDL is obtained per attribute. To deal with dependent and irrelevant attributes, we apply a structural improvement method that eliminates and/or joins attributes, based on mutual and conditional information measures.

We have applied this method for the analysis of colposcopy images for diagnosis of cervical cancer. For this we used the parameters from three mathematical models that characterize the temporal evolution of each pixel in the image. Each model has different number of parameters, in total 11 attributes, all continuous. There are three classes. Our method produces a very simple and efficient classifier with an accuracy of 95%.

J.S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140, pp. 622–631, 2006.

The rest of the document is organized as follows. Section 2 describes the Bayesian classifier and related work in structural improvement. Section 3 presents our method for model combination. Section 4 introduces the cervical cancer detection problem, and Section 5 the image processing and modeling stages. In the next section we present the experimental results. We conclude with a summary and directions for future work.

2 Related Work

There are several ways to combine several models for classification purposes. One is to build several classifiers, one for each model, and then combine the outputs of each classifier by, for example, majority voting [15]. Another is to build an ensemble or cascade of classifiers, as in Boosting [7]. We propose a different approach. We combine the parameters of all the models into a single classifier, and then use this for classification. We use is a simple, naive Bayes classifier. For building this classifier, we eliminate or join irrelevant and dependent attributes, through a *structural improvement* stage. Next, we briefly review the naive Bayes classifier and related work in structural improvement.

A Bayesian classifier obtains the posterior probability of each class, Ci, using Bayes rule, as the product of the prior probability of the class by the conditional probability of the attributes given the class (likelihood), divided by the probability of the attributes. The naive Bayes classifier (NBC) makes the simplifying assumption that the attributes are independent given the class, so the likelihood can be obtained by the product of the individual conditional probabilities of each attribute given the class. In this way, the number of parameters increases linearly with the number of attributes, instead of exponentially. Graphically, a NBC can be represented as star-structured Bayesian network [12], with a root node, C, that corresponds to the class variable, which is connected to the attributes, $E1, ..., En$. The attributes are assumed conditionally independent given the class, so there are not arcs between them. Thus, the posterior probability, $P(Ci \mid E1, ..., En)$, of class Ci is given by:

$$P(Ci \mid E1, ..., En) = P(Ci)P(E1 \mid Ci) \dots P(En \mid Ci)/P(E) \qquad (1)$$

2.1 Structural Improvement

The naive Bayes classifier assumes that the attribute are independent given the class. If this is not true, there are two basic alternatives. One is to transform the structure of the classifier to a Bayesian network, by introducing directed arcs between the dependent attributes. There are several variants of this approach which are described in [8]. The disadvantage is that the simplicity of the NBC is lost, so learning the model and then classifying new instances becomes more complex. The other alternative is to transform the structure but maintaining a star or tree-structured network. For this, [13] introduces 3 basic operations: (i) eliminate an attribute, (ii) join two attributes into a new combined variable,

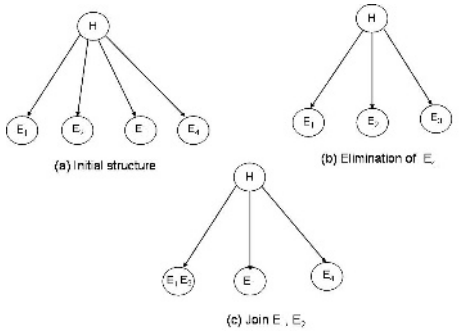


Fig. 1. Structural improvement: (a) original structure, (b) one attribute is eliminated, and (c) two attributes are joined into one variable

(iii) introduce a new attribute that makes two dependent attributes independent (hidden node). Figure 1 illustrates the first two operators, which are the ones we use in this work. These operations are based on statistical tests to measure the correlation of pairs of attributes given the class variable. Later, [11] proposes an alternative algorithm for variable elimination and merging (that correspond to the first two operators). The algorithm is based on two search procedures: (i) forward sequential selection and joining (FSSJ) and (ii) backward sequential elimination and joining (BSEJ). This procedures start from a full (empty) structure, and the they select attribute for elimination (addition) or for combination, testing the classification accuracy after each operation. The advantage of these approaches is that they preserve the simplicity and efficiency of the NBC.

In contrast to previous work, we combine elimination of irrelevant attributes with elimination or combination of dependent attributes. Also, our method is based in simple information measures, that avoid testing the classifier for each possible structure, and makes learning more efficient. Finally, our method also integrates a discretization stage to deal with continuous attributes.

3 Model Combination

The proposed method obtains an efficient Bayesian classifier for combining several models for classification. It considers (i) discretization of continuous variables, (ii) selection of relevant attributes, and (iii) elimination or combination of dependent attributes. The method obtains a naive Bayes classifier with the minimum number of attributes obtained form the different models. The general algorithm is the following: (1) Initialization, (2) Discretization, (3) Structural Improvement, and (4) Evaluation. Next we describe each stage in detail.

3.1 Initialization

This is step is done only once to build the initial classifier. It considers all the attributes from each model (full structure) and an initial partition for the

continuous attributes with two equal size intervals. Based on this initial structure, the parameters are learned from training data.

3.2 Discretization

Given the current structure, in this stage the discretization for each continuous attribute is optimized. The method is based on the minimum description length (MDL) principle. The MDL measure makes a compromise between the accuracy and complexity of the discretization. The measure we use, estimates the accuracy by measuring the mutual information between each attribute and the class; and the complexity by counting the number of parameters. This is done for all the continuous attributes. This stage is described in detail in [10].

3.3 Structural Improvement

Given the current discretization (the one with the best MDL from the previous stage), in this phase the structure is improved to eliminate superfluous attributes and eliminate or combine dependent attributes. This phase considers the following stages:

1. The mutual information between each attribute and the class are obtained, and those attributes that do not provide information (below a threshold) are eliminated.
2. The reminding attributes are tested using conditional mutual information (CMI) for each pair of attributes given the class. If this value is high for a pair of attributes it is an indication that the attributes are not independent, so these are candidates for elimination or combination.
3. Each pair of attributes with high CMI are considered for elimination or combination. One is deleted or both are combined into a single attribute. The option with better classification rate is selected.

This process is repeated until there are no more superfluous or dependent attributes. At this stage, the method can go back the discretization stage to optimize again the partitions for each continuous attribute, and then repeat structural improvement.

3.4 Evaluation

The final stage consists on evaluating the accuracy of the final classifier with test data.

4 Cervical Cancer Detection

Cervical cancer is the first cause of death in Mexican women. If it is detected early, the probability of cure is very high. After a Pap smear test, colposcopy is the most common technique to diagnose this disease because, although it

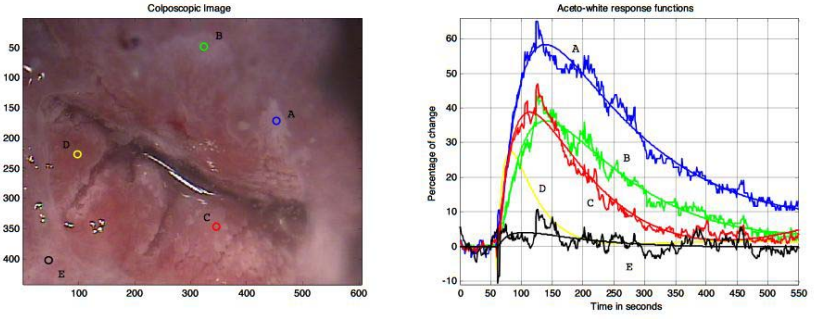


Fig. 2. An example of a colposcopic image, where five regions (A, B, C, D, E) were selected from the image sequence (left). The corresponding Acetowhite response functions are plotted against time (the X-axis is time and the Y-axis is the relative intensity), including the corresponding adjusted curve using model 2 (right).

is more expensive, it has a higher sensitivity and specificity [5]. Basically, the colposcopic test consists of the evaluation of the level of "white" color intensity that the cervical tissue reaches after acetic acid application. An example of a colposcopic image is shown in figure 2.

There are two problems to develop a colposcopic test. Firstly, the visual analysis has to be done by a well-trained gynecologist. Secondly, the evaluation of the images is subjective in the sense that there are not standard criterion to correlate the tissue color (whitening) with the lesion degree [6]. Many approaches have been proposed to automatically characterize cervical lesions from colposcopic images, most of them use color, texture, or shading, but none of these approaches have shown to be robust enough to be used as a diagnostic tool [14,9]. More recently, some researchers have suggested to use the temporal patterns intrinsic to the color changes, but there is not a complete understanding of how to represent the dynamics of the whitening that occurs after acetic acid application, and how to use it to make an automatic diagnosis adviser [2,4,3]. In previous work [1], we developed a technique to model the temporal evolution of the light changes in the tissue, and to parameterize the dynamics of these temporal changes using different mathematical models. In the following section, this technique is explained, as well as the different mathematical models developed. Then we present how the models are combined based on our methodology.

5 Image Processing and Modeling

5.1 Data Acquisition

Images were acquired using a colposcope (dfv Vasconsellos model CP-M7, with magnification 16 X without any optical filter) and a color camera (Sony SSC-DC50A), with a viewing distance of 20 cm. During the first minute of image acquisition, 60 images (640x480) were taken as base line reference (1 frame/second).

Then, after acetic acid application, five hundred and forty images were taken in 9 minutes using the same sampling frequency. In order to simplify image analysis, the images were processed in gray scale at a resolution of 74x99.

5.2 Image Analysis

The methodology used to analyze colposcopic images involves 3 main stages: (i) image Registration, (ii) time series construction, and (iii) modeling.

The acquisition of colposcopic images spans in average 9 minutes and even though that the patient is fixed, some small random movements are unavoidable, which are often local (patients breathing, movements due to the muscle tonus, etc.). To be able to analyze the sequence of images, i.e. compare and evaluate corresponding structures, the objects in the images should be brought into the same position by removing the differences due to the patient movements - the time series of colposcopic image has to be registered. In our case, we supposed the main source of the misalignments can be modeled by simple translation. The analysis of preliminary registration results showed that this assumption was correct. The method can transform the whole data using the same parameters, or can be local, depending on the local variations. It can be based directly on the image intensity values (area-based methods) or can be done using some features computed from the images (feature-based methods). Because colposcopic images do not contain many distinctive details, an area-based method was chosen.

The classical representative of area-based methods is the normalized cross-correlation. This method exploits matching directly image intensities. The measure of similarity is computed for window pairs from the input and reference images and its maximum is searched [16]. Because the tissue appearance changes over time due to the effect of the acetic acid effect, the searched pattern defined in the window can look different at a different times, so it is not appropriate to define a static reference image. The input and the reference images are updated continuously starting with the first and second images of the sequence respectively, then the input and the reference images are redefined by the second and the third images and so on. The starting points to initialize the search are updated by the last position in which the pattern window was found. This registration strategy allows not only to contend with the fact that the searched pattern changes over time, but also to reduce the spatial space over which to develop the search.

After the registration process, the signal to noise ratio was increased using a spatial low pass filter implemented using a kernel window (3x3). The intensity value of each pixel over time was used to construct a time series, we called this, the Aceto-white response functions (AwRFs). On a similar way, each AwRF was smoothed using a polynomial filter with a polynomial of order 2 and frames size 60. The filtered time series can be plotted against time (in the image sequence), some examples are shown in figure 2.

In previous work [1], three different mathematical models were used to model the evolution of each pixel and based on this to diagnose the probability of cancer.

Model 1 - Important Points. This model considers 3 important points in the curve:

1. Ts: the time to reach its maximum value.
2. Tb: the maximum value.
3. Tc: the time from the maximum value until the signal returns to its original value.

Model 2 - Polynomial. Polynomial of degree 5 that approximates the signal, with parameters, $P1, P2, P3, P4, P5$:

$$AwRF(t) = P1 + P2t + P3/t + P4/t^2 + P5/t^3 \tag{2}$$

For $0 < t < 1$, where $t(i) = sample_i/samples$. The model is a function of time, where the start point of the dynamic function is at time = 60 seconds (the time at which the acetic acid was applied). So if 600 images were acquired on the image sequence, $t(1)$ is equal to 0.1 (60/600). For analytical purposes the AwRFs can be divided in two sets: the upwards part and the downwards part. The first part represents that the signal reaches the maximum level at some time, and the second one represents the dynamics of how the signal goes back from the maximum to the base line. The peak time can be easily computed by solving $AwRF(t) = 0$ for t. In the same way, the maximum percentage of change reached can be known substituting for this t.

Model 3 - Parabola. Parabola approximation with parameters, Ph, Pk, Pp. After an analysis of the AwRFs we found that the first part of the signal can be approximated using a parabola. The standard form of the equation of a parabola with vertex (h, k) and directrix $y = k-p$ is expressed as $(x-h)2 = 4p(y-k)$ with respect to the vertical axis. Where the focus lies on the axis, p units from the vertex. Under this representation the percentage of change and the time peak can be seen as the vertex (h, k) of the parabola, and p as an estimator of the speed of change. An example of the parabolic approximation for different curves is depicted in figure 3.

6 Experimental Results

We evaluated our methodology in the classification of different regions of colposcopy videos. Firstly, some image sequences are classified with the help of an expert. Secondly, based on the training cases, the time series are obtained and described using the 3 models presented in the previous section. Finally, the model combination methodology is applied and a Bayesian classifier is generated that combines the 3 models. This combined classifier was tested with other image sequences, different to the ones for training.

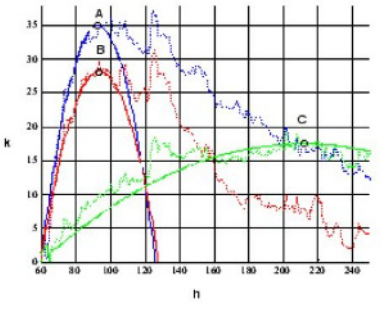


Fig. 3. Some examples of parabolic approximations to typical AwRFs

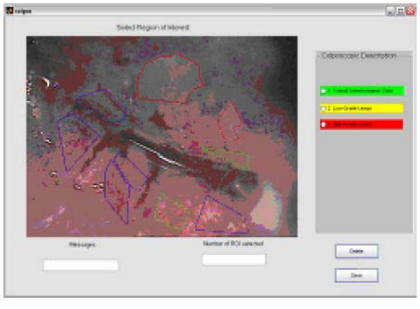


Fig. 4. An example of a colposcopic image with the interface used by the expert to label the training cases

6.1 Knowledge Acquisition

The knowledge extraction process was implemented asking the colposcopist to make an analysis of the image sequence. Assisted by a graphical interface in which a representative image of the cervix was shown, the colposcopist was asked to define over the image the different regions with regards to three categories usually reported in a colposcopic description: *typical transformation zone*, *low-grade lesion*, and *high-grade lesion*. The main idea is to ask the expert to make a colposcopic evaluation of one original image and to define regions with an associated label which correspond to a colposcopic feature. This knowledge extraction procedure is made through the use of a graphical interface depicted in figure 4.

6.2 Results

We applied our methodology by considering the 3 models, so all the parameters are the attributes for the classifier, 11 continuous attributes. There are three classes: normal (T), low degree (A) and high degree (B). For the experiments

we used 1055 sample data labeled by an expert. We used a holdout testing procedure, with approx. 2/3 of the data for training and 1/3 for testing (800 for training and 255 for testing).

After obtaining the best discretization for each attribute, the results of the structural improvement stage are summarized in table 1. We can observe that the algorithm eliminates initially a number of redundant attributes, then it combines two dependent attributes and then it eliminates two more attributes, until it arrives at the final structure with 4 attributes: $Ts - Pk, Tc, P2, P4$, where one of them is the combination of two attributes of the original models. This final classifier obtains a 95% accuracy. In comparison, the results with the best single model were of aprox. 90% accuracy, using the same data and testing procedure. We also compared it with other classifiers using the same data (using all the attributes): (i) naive Bayes, 89%, (ii) tree augmented naive Bayes (TAN), 94%, and (iii) decision tree (C4.5), 94%. The accuracy of our best model is clearly superior to the Naive Bayes and similar to TAN and C4.5. However, the model obtained is much simpler than TAN and C4.5, and thus more efficient.

7 Conclusions and Future Work

We have developed a methodology to combine several models using a Bayesian approach. The method selects the most relevant attributes from several models, and produces a Bayesian classifier that has a high accuracy and at the same time is very efficient. Based on conditional information measures, the method eliminates irrelevant variables, and joins or eliminates dependent variables; until an *optimal* Bayesian classifier is obtained. We have applied this method for diagnosis of precursor lesions of cervical cancer. Based on the dynamics of a video sequence of colposcopy images, 3 different mathematical models were generated to describe the Aceto-white response functions over time. By combining these 3 models with our approach, we generated a Bayesian classifier with 4 attributes (one is a combination of 2 of the original parameters) that produces a 95% accuracy for the test cases.

We are working on extending this method for dynamic models, based on dynamic Bayesian networks. In the future we will like to apply the methodology

Table 1. Structural improvement stage for cancer diagnosis

Operation	No. attr.	attributes	Acc.
	11	TsTbTcP1P2P3P4P5PhPkPp	88
Join Ts-Pk	10	Ts-PkTbTcP1P2P3P4P5PhPp	89
Elim. P1	9	Ts-PkTbTcP2P3P4P5PhPp	89
Elim. Pp	8	Ts-PkTbTcP2P3P4P5Ph	90
Elim. Ph	7	Ts-PkTbTcP2P3P4P5	90
Elim. Tc	6	Ts-PkTbTcP2P4P5	91
Elim. P3	5	Ts-PkTcP2P4P5	92
Elim. P5	4	Ts-PkTcP2P4P5	95

in other domains, considering more models and parameters, to test the scalability of this approach.

References

1. H. Acosta-Mesa, B. Zitová, H. Ríos-Figueroa, N. Cruz-Ramírez, A. Marín-Hernández, R. Hernández-Jiménez, B. Cocotle-Ronzón, and E. Hernández-Galicia. Cervical cancer detection using colposcopic images: a temporal approach. In *Proc. Sixth International Conference on Computer Science (ENC'05)*, pages 158–164. IEEE, 2005.
2. M. Anderson, A. M. Jordan J., and F. Sharp. *A Text and Atlas of Integrated Colposcopy.* Mosby, 1993.
3. W. Pogue B, A. Zelenchuk, W. Harper, G. C. Burke, E. E. Burke, and D. M. Harper. Analysis of acetic acid-induced whitening of high-grade squamous intraepithelial lesions. *Journal of Biomedical Optics*, pages 397–403, 2001.
4. C. Balas. A novel optical imaging method for the early detection, quantitative grading, and mapping of cancerous and precancerous lesions of cervix. *IEEE Transactions on medical imaging*, pages 96–104, 2001.
5. E. Burghardt, P. Hellmuth, and F. Girardi. *Primary care colposcopy.* Thieme, 2004.
6. B. L. Craine, C. J. O'Toole, and Q. Ji. Digital imaging colposcopy: Corrected area measurements using shape-from-shading. *IEEE Transactions on medical imaging*, pages 1003–1010, 1998.
7. R. Freund and E. Schapire. A short introduction to boosting. *Journal of the Japanese Society for Artificial Intelligence*, 14:771–780, 1999.
8. N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.
9. Q. Ji and Eric Craine. Texture analysis for classification of cervix lesions. *IEEE Transactions on medical imaging*, pages 1144–1149, 2000.
10. M. Martínez and L.E. Sucar. Learning an optimal naive bayes classifier. In *Proc. Congreso de Investigación y Extensión.* ITESM, 2006.
11. Michael J. Pazzani. Searching for atribute dependencies in bayesian classifiers. *In preliminary Papers of the Intelligence and Statistics*, pages 424–429, 1996.
12. J. Pearl. *Probabilistic reasoning in intelligent systems.* Morgan Kaufmann, Palo Alto, Calif., U.S.A., 1988.
13. L.E. Sucar, D.F. Gillies, and D.A. Gillies. Objective probabilities in expert systems. *Artificial Intelligence*, 61:187–208, 1993.
14. A. T. Vlastos, Andres Zuluaga, and Michele Follen. New approaches to cervical cancer screening. *Contemporany Ob/Gyn*, pages 87–103, 2002.
15. X. Yi, Z. Kuo, and C. Zhang. Classifier combination based on active learning. In *Proc. of the 17th International Conference on Pattern Recognition (ICPR'04)*, pages 184–187. IEEE, 2004.
16. B. Zitova. Image registration methods: a survey. *Image and Vision Computing*, pages 977–1000, 2003.

Author Index