Hoai An Le Thi
Pascal Bouvry
Tao Pham Dinh (Eds.)

Communications in Computer and Information Science 14

Modelling, Computation and Optimization in Information Systems and Management Sciences

Second International Conference MCO 2008
Metz, France - Luxembourg, September 2008
Proceedings

Springer

Communications
in Computer and Information Science 14

Hoai An Le Thi Pascal Bouvry
Tao Pham Dinh (Eds.)

Modelling, Computation and Optimization in Information Systems and Management Sciences

Second International Conference MCO 2008
Metz, France – Luxembourg, September 8-10, 2008
Proceedings

Springer

Volume Editors

LE THI Hoai An
Laboratory of Theoretical and Applied Computer Science
UFR MIM, Paul Verlaine University of Metz
Metz, France
E-mail: lethi@univ-metz.fr

Pascal Bouvry
Faculty of Sciences, Technology and Communications
University of Luxembourg
Luxembourg
E-mail: pascal.bouvry@uni.lu

PHAM DINH Tao
Laboratory of Mathematics
National Institute for Applied Sciences - Rouen
Mont Saint Aignan, France
E-mail: pham@insa-rouen.fr

Library of Congress Control Number: 2008934316

CR Subject Classification (1998): C.0, K.6, C.4, I.5, H.1

ISSN 1865-0929
ISBN-10 3-540-87476-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-87476-8 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12514144 06/3180 5 4 3 2 1 0

Preface

This volume contains the 65 selected full papers (from 160 submitted ones) presented at the MCO 2008 conference, held on September 8–10, 2008 at Paul Verlaine University of Metz, France and the University of Luxembourg.

MCO 2008 was the second event in the series of MCO conferences on *Modelling, Computation and Optimization in Information Systems and Management Sciences* organized by LITA, the Laboratory of Theoretical and Applied Computer Science, University Paul Verlaine, Metz. Now recognized as a high-quality international conference, MCO takes place in Metz every four years. The first conference, MCO 2004, brought together 100 scientists from 21 countries and was a great success. It included 8 invited plenary speakers, 70 papers presented and published in the proceedings, "Modelling, Computation and Optimization in Information Systems and Management Sciences", edited by Thi Hoai An and Pham Dinh Tao, Hermes Sciences Publishing, June 2004, 668 pages, and 22 papers published in the European Journal of Operational Research and in the Journal of Global Optimization. This time the Computer Science and Communications Research Unit, University of Luxembourg joined forces with LITA in the organization of the MCO conference.

MCO 2008 covered several fields of Management Science and Information Systems: Computer Sciences, Information Technology, Mathematical Programming, and Optimization and Operations Research, through the five main topic areas: Optimization and Decision Making; Data Mining Theory, Systems and Applications; Computer Vision and Image Processing; Computer Communications and Networks; and Optimization and Search Techniques for Security, Reliability, Trust. It allowed researchers and practitioners to clarify the recent developments in models and solutions for decision making in Engineering and Information Systems and to interact and discuss how to reinforce the role of these fields in potential applications of great impact.

Continuing the success of the first conference, MCO 2004, MCO 2008 brought together 6 invited plenary speakers and more than 120 scientists from 27 countries. The scientific program consisted of 6 plenary lectures and of the oral presentation of 65 selected full papers as well as 34 selected abstracts covering all main topic areas.

We would like to thank all those who contributed to the success of the conference and to this book of proceedings. In particular we would like to mention the authors as well as the members of the scientific committee and the referees for their efforts and cooperation. Finally, the interest of the sponsors in the meeting and their assistance are gratefully acknowledged.

June 2008

Le Thi Hoai An
Bouvry Pascal
Pham Dinh Tao

Organization

MCO 2008 was organized by the Laboratory of Theoretical and Applied Computer Science, Paul Verlaine University of Metz, France, in collaboration with the Computer Science and Communications Research Unit, University of Luxembourg.

Organizing Committee

Conference Chair	Le Thi, Hoai An (LITA – Paul Verlaine University of Metz, France)
Conference Co-chair	Bouvry, Pascal (University of Luxembourg, Luxembourg)

Members

Aignel, Damien	LITA – Paul Verlaine University of Metz, France
Boudjeloud, Lydia	LITA – Paul Verlaine University of Metz, France
Brucker, François	LITA – Paul Verlaine University of Metz, France
Conan-Guez, Brieu	LITA – University Paul Verlaine-Metz, France
Gély, Alain	LITA – Paul Verlaine University of Metzz, France
Le Hoai, Minh	LITA – Paul Verlaine University of Metz, France
Nguyen, Quang Thuan	LITA – Paul Verlaine University of Metz, France
Schleich, Julien	LITA – Paul Verlaine University of Metz, France
Veneziano, Thomas	LITA – Paul Verlaine University of Metz, France

Program Committee

Program Chair	Le Thi, Hoai An (LITA – Paul Verlaine University of Metz, France)
Program Co-chair	Pascal, Bouvry (University of Luxembourg, Luxembourg)
	Pham, Dinh Tao (INSA-Rouen, France)

Members

El-Houssaine Aghezzaf	University of Gent, Belgium
Enrique Alba	University of Malaga, Spain
Azeddine Beghadi	University of Paris 13, France
Alain Billionet	CNAM, Paris, France
Raymond Bisdorff	University of Luxembourg, France
Hans-Hermann Bock	RWTH Aachen University, Germany
Alexander Bockmayr	University of Berlin, Germany
Serge Chaumette	University of Bordeaux, France
John Clark	University of York, UK
Ali Mohammad-Djafari	CNRS-Supelec, France
Arnaud Freville	Université de Valenciennes, France
Gerard Govaert	University of Compiegne, France
Frédéric Guinand	University of Le Havre, France
Nalan Gulpinar	University of Warwick, UK
Pierre Hansen	University of Montreal, Canada
Jin-Kao Hao	University of Angers, France
Pham Hoang	Rutgers University, France
Luc Hogie	INRIA Sophia-Antipolis, France
Joaquim Judice	University of Coimbra, Portugal
Djamel Khadraoui	CRP H. Tudor, Luxembourg
Le Ngoc Tho	McGill University, Canada
Abdel Lisser	Université Paris Sud, France
Nadine Meskens	FUCaM, Belgium
Amédéo Napoli	LORIA, France
Panos Pardalos	University of Florida, USA
Gerard Plateau	University of Paris 3, France
Jean Marie Proth	INRIA, France
Nidhal Rezg	Paul Verlaine University of Metz, France
Berc Rustem	Imperial College, UK
Marc Schoenauer	INRIA, France
Christoph Schnoerr	University of Mannheim, Germany
Franciszek Seredynski	Polish Academy of Science, Warsaw, Poland
Gilles Venturini	University of Tours, France
Gerhard-Wilhelm Weber	Middle East Technical University, Turkey
Henry Wolkowicz	University of Waterloo, Canada
Zhijun Wu	University of Iowa, USA
Adnan Yassine	University of Le Havre, France
Yinyu Ye	Stanford University, USA

External Referees

N. Belkhiter	D. Caragea	B. Dorronsoro
G. Behre	B. Conan-Guez	R. El-Assoudi
F. Brucker	G. Danoy	A. Gély

C. Gout
A. Knippel
A. Mazeika
R. Mazza
D.Y. Nguyen
T.K. Phan
D. Singer
A. Unwin
S. Varette

Sponsoring Institutions

Université Paul Verlaine de Metz (UPV-M)
Laboratoire d'Informatique Théorique et Appliquée, UPV-M
UFR Mathématique Informatique Mécanique Automatique, UPV-M
Université du Luxembourg (UL)
Fonds National de la Recherche du Luxembourg
Computer Science and Communications Research Unit, UL
Conseil Général de la Moselle
Conseil Régional de Lorraine

Table of Contents

MCO 2008

Optimization and Decision Making

Data Mining Theory, Systems and Applications

Computer Vision and Image Processing

Computer Communications and Networks

Optimization and Search Techniques for Security, Reliability, Trust

Optimal Flight Paths Reducing the Aircraft Noise during Landing

Lina Abdallah

MAPMO - UMR 6628
Universite d'Orléans - BP 6759
45067 Orléans cedex 02 France
lina.abdallah@univ-orleans.fr

Abstract. This study concerns the development and the resolution of an optimization model minimizing the aircraft noise levels around the airports. Our problem is stated as a nonconvex and nonlinear control problem. We used two numerical approaches : direct and indirect. Some numerical results of these approaches are presented and discussed in this paper.

Keywords: Optimization, optimal control, aircraft noise abatement.

1 Introduction

For many years, commercial aircraft noise represented a serious social and environmental problem for the population living near the airports. A possible solution to decrease aircraft noise is the practicability of new flight paths with the best management of operational procedures. The objective is to make these paths practical where the predominant task is to maintain high safety flight during landing operations.

In this context, we developed and solved an optimization model, minimizing the noise level. To this end, two stages were needed. Firstly, we defined the components which formulated an ODE optimal control problem: the ODE system represents the flight dynamics of the aircraft in the vertical plane, the constraints concern some flight safety and comfort requirements, and the cost function is an aircraft noise index describing the effective noise level of the noise aircraft event.

In the second stage, we used two numerical approaches : direct and indirect. The direct approach is based on the Karush-Kuhn-Tucker, where we discretized the problem by partitioning the time interval and we reduced the optimal control problem into finite-dimensional, then the large nonlinear program is solved by a standard NLP solver. The indirect approach is based on Pontryaguine's principle and an adapted interior point algorithm. Basically, we used the primal-dual formulation of the optimality conditions. A discrestization of these conditions transforms the system to a set of non-linear equations that can be solved according to the discretized variables. For the implementation, we used the AMPL [1] language and the SNOPT solver [8].

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 1–10, 2008.

This paper is organized as follow : a first section presenting the aim of the work; a second section, which present the details of the optimal control problem; a third section, which discuss the developed methods to solve the problem; a fourth section which is devoted to numerical experiments and finally a conclusion.

2 Optimal Control Problem

2.1 Dynamics of Problem

The system of differential equations commonly employed in aircraft trajectory analysis is the following six-dimension system derived at the center of mass of the aircraft [3]:

$$(ED) \qquad \begin{cases} \dot{V} = g\left(\dfrac{T\cos\alpha - D}{mg} - \sin\gamma\right) \\ \dot{\gamma} = \dfrac{1}{mV}\left((T\sin\alpha + L)\cos\mu - mg\cos\gamma\right) \\ \dot{\chi} = \dfrac{(T\sin\alpha + L)\sin\mu}{mV\cos\gamma} \\ \dot{x} = V\cos\gamma\cos\chi \\ \dot{y} = V\cos\gamma\sin\chi \\ \dot{h} = V\sin\gamma \end{cases}$$

where V, γ, χ, α and μ are respectively the speed, the angle of descent, the yaw angle, the angle of attack and the roll angle. The position of the aircraft is (x, y, h).

The variables T, D, L, m and g are respectively the engine thrust [7], the drag force, the lift force, the aircraft mass [3] and the aircraft weight acceleration. Those variables are expressed as:

$$\begin{cases} T = T_0\delta_x\dfrac{\rho}{\rho_0}(1 - M + \dfrac{M^2}{2}) \\ L = \frac{1}{2}\rho S V^2 C_{z_\alpha}\alpha \\ D = \frac{1}{2}\rho S V^2\left[C_{x_0} + k_i C_{z_\alpha}^2\alpha^2\right] \\ \rho = \rho_0(1 - 22.6\times 10^{-6}h)^{4.26} \\ c = 10.1\sqrt{273 - 0.0065h} \\ M = \dfrac{V}{c} \end{cases}$$

With T_0 the full thrust, ρ the density of air at altitude h, δ_x the throttle setting, ρ_0 the atmospheric density at the ground ($= 1.225\ kg/m^3$) and c the speed of sound at altitude h.

The previous system of equations (ED) respects the assumptions of a constant weight, symmetric flight and constant gravitational attraction.

Vertical Plan. We restrict our study of the flight path optimization in the vertical plane. The equations in the vertical plane are obtained by setting the following conditions :

1. $y = cste$
2. Yaw angle and roll angle, $\chi = \mu = 0$

In this case (ED) becomes:

$$\begin{cases} \dot{x} = V\cos\gamma \\ \dot{h} = V\sin\gamma \\ \dot{V} = \dfrac{1}{m}(T\cos\alpha - D) - g\sin\gamma \\ \\ \dot{\gamma} = \dfrac{1}{mV}(T\sin\alpha + L) - \dfrac{g\cos\gamma}{V} \end{cases} \tag{1}$$

The system of equations (1) can be written in the following matrix form:

$$\dot{z}(t) = f(z(t), u(t))$$

where

$$\begin{aligned} z : [t_0, t_f] &\longrightarrow I\!R^4 \\ t &\longrightarrow z(t) = [V(t), \gamma(t), x(t), h(t)] \text{ is the state variable,} \end{aligned}$$

$$\begin{aligned} u : [t_0, t_f] &\longrightarrow I\!R^2 \\ t &\longrightarrow u(t) = [\alpha(t), \delta_x(t)] \text{ is the control variable,} \end{aligned}$$

t_0 and t_f are the initial and final times.

2.2 Path Constraints

Along the trajectory, we have some safety requirements and comfort constraints. For that, we have to respect:

$$\begin{aligned} 1.3V_{s_0} &\leq V \leq V_{max} \\ \delta_{x_{min}} &\leq \delta_x \leq \delta_{x_{max}} \\ \gamma_{min} &\leq \gamma \leq \gamma_{max} \\ \alpha_{min} &\leq \alpha \leq \alpha_{max} \end{aligned}$$

where V_{s_0} represents the stall velocity, i.e. the limited velocity at which the aircraft can produce enough lift to balance the aircraft weight.

These inequalities can be put into the following form:

$$a \leq C(z(t), u(t)) \leq b$$

where a and b are two constant vectors.

2.3 Cost Function

The cost function may be chosen as any of the usual aircraft noise indices, which describes the effective noise level of the aircraft noise event [12,13], like SEL (Sound Exposure Level), the $EPNL$ (Effective Perceived Noise Levels) or the $L_{eq,\Delta t}$ (Equivalent noise level)...

This study is limited to the sound effects observed during a given interval of time, thus we choose to minimize the index $L_{eq,\Delta T}$ during landing. It is expressed as:

$$L_{eq,\Delta T} = 10 \log \frac{1}{\Delta T} \int_{t_0}^{t_f} 10^{0.1 L_P(t)} dt \tag{2}$$

where t_0, t_f and $L_P(t)$ are respectively, the initial time, the final time and the overall sound pressure level (expressed in decibels (dB)). ΔT is equal to $t_f - t_0$. The analytic formula to compute the noise levels at any reception point is:

$$L_P = L_{ref} - 20 \log_{10} R + \Delta_{atm} + \Delta_{ground} + \Delta_V + \Delta_f \tag{3}$$

where L_{ref} is the sound level at the source, $20 \log_{10} R$ is a correction factor due to geometric divergence, Δ_{atm} is the attenuation due to atmospheric absorption of sound. The other terms Δ_{ground}, Δ_V and Δ_f correspond respectively to the ground effects, correction for the Doppler and correction for the frequency.

In this study, we have used a semi-empirical model to predict noise that is generated by conventional-velocity-profile jets emitted from coaxial nozzles. This model can be used to express the jet noise [9] which corresponds to the main predominated noisy source, under the following form:

$$L_P(t) = \begin{cases} 141 + 10 \log_{10} \left(\dfrac{\rho_1}{\rho} \right)^w + 10 \log_{10} \left(\dfrac{V_e}{c} \right)^{7.5} + 10 \log_{10} s_1 \\ \\ +10 \log_{10} \left(\left(1 - \dfrac{v_2}{v_1} \right)^{me} + 1.2 \dfrac{\left(1 + \dfrac{s_2 v_2^2}{s_1 v_1^2} \right)^4}{\left(1 + \dfrac{s_2}{s_1} \right)^3} \right) \\ \\ +3 \log_{10} \left(\dfrac{2 s_1}{\pi d^2} + 0.5 \right) + \log_{10} \dfrac{\tau_1}{\tau_2} - 20 \log_{10} R \\ \\ -15 \log_{10} (C_D(M_c, \theta)) - 10 \log_{10} (1 - M \cos \theta), \end{cases} \tag{4}$$

where

$$\begin{cases} V_e & = v_1 [1 - (V/v_1)]^{2/3} \\ C_D(M_c, \theta) & = \left[(1 + M_c \cos \theta)^2 + 0.04 M_c^2 \right] \\ M_c & = 0.62 (v_1 - V)/c \\ \rho & = \rho_0 (1 - 22.6 \times 10^{-6} h)^{4.26} \\ w & = \dfrac{3 (V_e/c)^{3.5}}{0.6 + (V_e/c)^{3.5}} - 1. \end{cases}$$

Here, $\theta,\ v,\ s$ and τ represent respectively the directivity angle, the speed of jet gas, the area of coaxial engine nozzle and the temperature. The subscripts 1, 2 correspond to the inner and outer contours.

The distance R between source and observer is

$$R = l^2 + h^2(\cos(\gamma))^2$$

where l the lateral distance, γ the angle of descent and h the altitude.

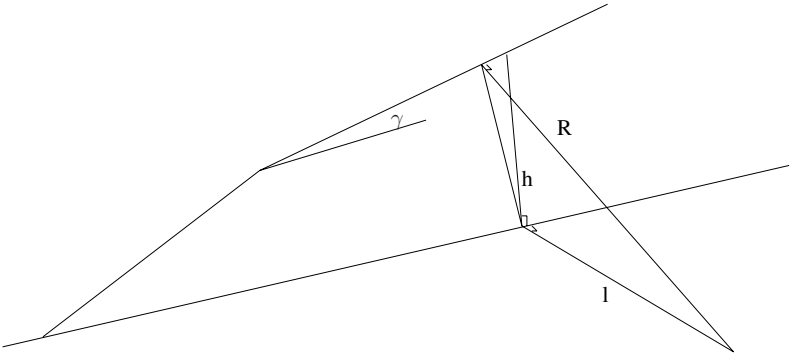


Fig. 1. Distance R between the source and observer

Taking into account the formulas (4) and (2), we obtain our cost function in the following integral function form

$$\begin{aligned} J : \mathcal{C}^1([t_0, t_f], I\!R^4) \times \mathcal{C}^1([t_0, t_f], I\!R^2) &\longrightarrow I\!R \\ (z(t), u(t)) &\longrightarrow J(z,u) = \int_{t_0}^{t_f} \ell(z(t), u(t))dt. \end{aligned}$$

J is the criterion for the noise level.

Finding an optimal trajectory, minimizing the noise level during landing, is a mathematical problem that can be stated as an optimal control problem as follow:

$$(OCP) \quad \begin{cases} \min\ J(z,u) = \displaystyle\int_0^{t_f} \ell(z(t), u(t))dt \\ \dot{z}(t) = f(z(t), u(t)), \forall t \in [0, t_f] \\ z_{I_1}(0) = c_1,\ z_{I_2}(t_f) = c_2 \\ a \leq C(z(t), u(t)) \leq b \end{cases}$$

where $J : I\!R^{n+m} \to I\!R, f : I\!R^{n+m} \to I\!R^n$ and $C : I\!R^{n+m} \to I\!R^q$ correspond respectively to the cost function, the dynamic of the problem and the constraints. The initial and final values for the sate variables $(h(0), V(0))$ and $h(t_f)$ are fixed.

3 Methods of Resolution

Numerical methods for solving control problems governed by ordinary differential equations fall into two categories, the indirect methods and the direct approach [2,10]. In this paper, we present the two approaches for solving the problem OCP.

3.1 Indirect Method

We set $\mathcal{H} : I\!R^n \times I\!R^m \times I\!R^n \times I\!R^q \times I\!R^q \longrightarrow I\!R$ the hamiltonian function of the problem OCP:

$$\mathcal{H}(z,u,p,\lambda,\mu) = \ell(z,u) + p^t f(z,u) + \lambda^t(C(z,u)-a) + \mu^t(b - C(z,u))$$

where λ, μ are the multiplicators associated to the constraints and p is the costate vector.

We describe now the optimality conditions (OC) for the OCP problem:

$$(OC) \quad \begin{cases} \dot{z}(t) = f(z(t),u(t)) \\ \dot{p}(t) = -\mathcal{H}_z(z(t),u(t),p(t),\lambda(t),\mu(t)) \\ u(t) = Argmin_w \mathcal{H}(z(t),w,p(t),\lambda(t),\mu(t)) \\ 0 \quad = \lambda.(C(z(t),u(t)-a), \quad \lambda \geq 0 \\ 0 \quad = \mu.(b - C(z(t),u(t)), \quad \mu \leq 0 \end{cases}$$

It is difficult to solve numerically the last two equations. In order to avoid this kind of problem, the interior point method could be used. This method consists of perturbing by a positif parameter the complementary conditions, then we obtain the following system:

$$(OC_\varepsilon) \quad \begin{cases} \dot{z}(t) \ = f(z(t),u(t)) \\ \dot{p}(t) \ = -\mathcal{H}_z(z(t),u(t),p(t),\lambda(t),\mu(t)) \\ u(t) \ = Argmin_w \mathcal{H}(z(t),w,p(t),\lambda(t),\mu(t)) \\ \mathbf{1}\varepsilon \ = \lambda.(C(z(t),u(t)) - a), \quad \lambda \geq 0 \\ -\mathbf{1}\varepsilon = \mu.(b - C(z(t),u(t))), \quad \mu \leq 0 \end{cases}$$

To solve OC, we have to solve a sequence of problems OC_ε by tending ε to zero. When ε decrease to 0, the solution of optimal conditions OC_ε is a solution for OC.

Discretization of Optimal Conditions. To compute the solution of the continuous optimal conditions, we first discretize them. We obtain a set of non-linear equations, which has to be solved for the discretized control, state and costate vectors using a Newton method. For the discretization, we choose an Euler schema. The discretization of the optimal conditions OC_ε gives the following system:

$$\begin{cases} z_{k+1} = z_k + \mathrm{h} f(u_k,z_k), \quad k = 0,\ldots,N-1 \\ p_{k+1} = p_k - \mathrm{h}\mathcal{H}_z(z_k,u_k,p_k,\lambda_k,\mu_k), \quad k = 0,\ldots,N-1 \\ 0 \quad = \mathcal{H}_u(u_k,z_k,p_k,\lambda_k,\mu_k), \quad k = 0,\ldots,N \\ \mathbf{1}\varepsilon \ = \lambda_k.(C(z_k,u_k) - a), \quad \lambda_k \geq 0, \quad k = 0,\ldots,N \\ -\mathbf{1}\varepsilon = \mu_k.(b - C(z_k,u_k)), \quad \mu_k \leq 0, \quad k = 0,\ldots,N \end{cases}$$

Finally we have a large set of equations to be solved under the boundary constraints corresponding to the multiplicators.

$$(N_\varepsilon) \quad \begin{cases} F_\varepsilon(X) = 0 \\ \lambda_k \geq 0 \\ \mu_k \leq 0 \end{cases}$$

where F_ε is the set of optimal conditions, and X the variable vector $X = (z_k, u_k, p_k, \lambda_k, \mu_k)$.

3.2 Direct Method

We discretize the control and the state with identical grid and reduce the optimal control problem into finite-dimensional, then the large nonlinear program is solved by a standard NLP solver.

We use an equidistant discretization of the time interval as

$$t_k = t_0 + k\mathrm{h}, \quad k = 0, ..., N \quad \text{and} \quad \mathrm{h} = \frac{t_f - t_0}{N}.$$

Then we consider that $u(.)$ is parameterized as a piecewise constant function:

$$u(t) := u_k \ \text{ for } \ t \in [t_{k-1}, t_k]$$

and use an Euler scheme to discretize the dynamic:

$$z_{k+1} = z_k + \mathrm{h} f(z_k, u_k), \quad k = 0, \ldots, N-1.$$

The new cost function is stated as:

$$\sum_{k=0}^{N} \ell(z_k, u_k).$$

The continuous problem is replaced by the following discretized control problem:

$$(NLP) \quad \begin{cases} \min\limits_{(z_k, u_k)} \sum\limits_{k=0}^{N} \ell(z_k, u_k) \\ z_{k+1} = z_k + \mathrm{h} f(z_k, u_k), \quad k = 0, \ldots, N-1 \\ z_{0_{I_1}} = c_1, \ z_{N_{I_2}} = c_2 \\ a \leq C(z_k, u_k) \leq b, \quad k = 0, \ldots, N \end{cases}$$

4 Numerical Application

We consider an aircraft landing in the vertical plane with initial conditions $h_0 = 3500\,m, V_0 = 180\,m/s$ and a final condition $h_f = 500\,m$. The landing time is fixed to $t_f = 10$ min.

We present here solutions obtained with the two considered approaches. The problem OCP is discretized along its state (h, V, γ) and control (α, δ_x). To solve NLP and N_ε, we developed an AMPL [1] model and used a robust solver SNOPT [8]. We have chosen this NLP solver after numerous comparisons with some other standard solvers available on the NEOS (Server for Optimization) platform.

4.1 Indirect Method

We solve a sequence of problems N_ε (tending ε to zero). We initialize the problem N_ε, by centering the sate and control. Then we initialize the Lagrange multiplicators as follow:

$$\lambda = \varepsilon(C(z,u) - a)^{-1}, \quad \mu = \varepsilon(b - C(z,u))^{-1}.$$

For the implementation of the penalty parameter ε, many strategies exist in the literature [6]. We used the following strategy:

$$\varepsilon_{k+1} = \varepsilon_k / 5.$$

We present the value of the noise in function of ε in the second column of the table 1. The third column gives the measurement of the feasibility error, the last column summarizes the exit message obtained with the SNOPT solver [8].

Table 1. The obtained solution in function of ε

ε	Noise (ε)	Feasible	Exit
1	63.3	$5.6e-12$	Optimal solution
0.2	62.7	$4.0e-12$	Optimal solution
0.04	62.3	$7.3e-14$	Optimal solution
0.008	62.2	$8.8e-12$	Optimal solution
0.0016	62.1	$1.1e-11$	Optimal solution
0.000032	62.1	$2.3e-07$	Optimal solution
$6.4e-05$	62.1	$2.5e-07$	Optimal solution
$1.28e-05$	62.1	$2.1e-07$	Optimal solution

For each iteration of interior point method, the algorithm (SNOPT [8]) found a solution with a very high accuracy.

Figure 2 shows the solution trajectory, control strategy and noise evolution.

We clearly see that the control δ_x is always saturated when the angle of attack α is not. The state variable γ is bang-bang between its prescribed bounds. The noise level is highest at the end of the trajectory and the altitude h plays a predominant role in the noise level.

4.2 Direct Method

Once OCP is transformed into a Non Linear Programming NLP problem, we use a standard NLP solver SNOPT and get a very good accuracy and an optimal solution. The obtained noise is equal to 62 dB and the feasibility error is equal to $1.8e-09$. Figure 3 shows the solution trajectory, control strategy and noise evolution.

We obtain approximately the same characteristic for the trajectory and for the noise level (~ 62 dB) from both approaches and confirm that the solution is optimal.

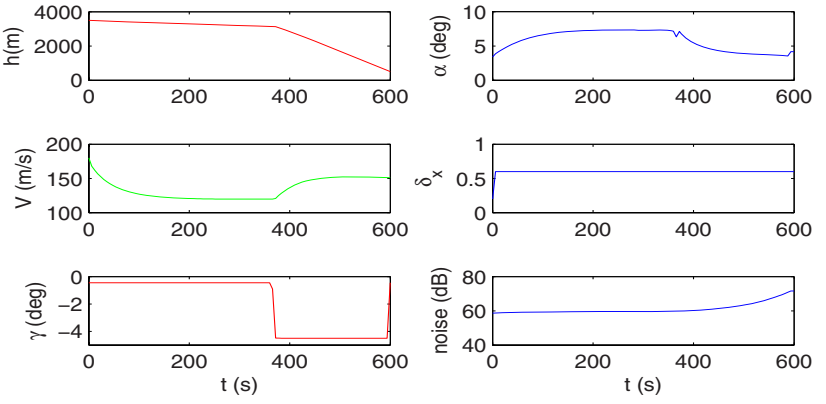


Fig. 2. Solution of N_ε for 100 discretization points

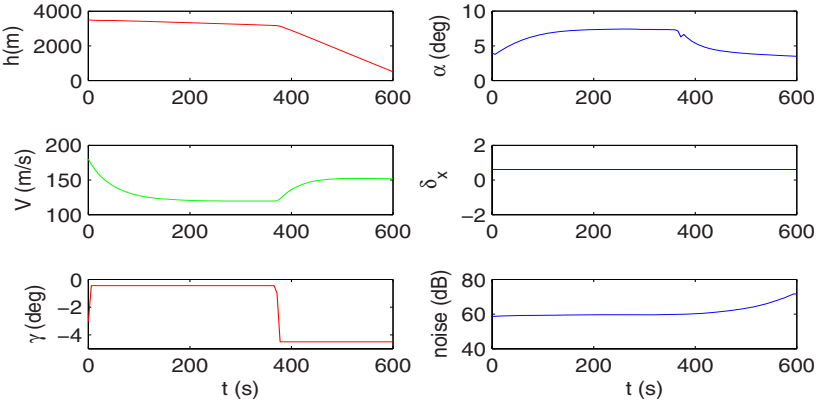


Fig. 3. Solution of NLP for 100 discretization points

5 Conclusion

This paper presents preliminary work for generating flight paths that minimize aircraft noise levels.

We have performed a numerical computation for indirect and direct approaches. An optimal solution is found in both cases, with a high accuracy. The two approaches give the same solution for the noise level and for the trajectory characteristic and confirm that the obtained solution is optimal. The direct approach is easier to implement.

This study is restricted to the vertical plane, an extension of the analysis should consider the problem in the space by using the general nonlinear system and by fixing one or several observers in the ground.

Acknowledgments. We would like to thank Mounir Haddou for his valuable help for computational aspects and Salah Khardi for his helpful discussions.

References

1. AMPL: A Modeling Language for Mathematical Programming, http://www.ampl.com
2. Berend, N., Bonnans, F., Haddou, M., Varin, J., Talbot, C.: An Interior-Point Approach to trajectory Optimization, INRIA Report, N. 5613 (2005)
3. Boiffier, J.: The Dynamics of Flight. John Wiley and Sons, Chichester (1998)
4. Bonnans, J.F., Gilbert, J.C., Lemarechal, C., Sagastizabal, C.: Optimisation Numrique: aspects et thoriques, Mathmatiques et Applications. Springer, Heidelberg (1997)
5. Bonnans, J.F., Launay, G.: Large-scale Direct Optimal Control Applied to a Re-Entry Problem. J. of Guidance. Control And Dynamic 21(6) (1998)
6. Dussault, J.P., Elafia, A.: On the superlinear convergence order of the logarithmic barrier algorithm (1999)
7. Mattingly, J.D.: Elements of Gas Turbine Propulsion. McGraw-Hill, New York (1996)
8. Gill, P., Murray, W., Saunders, M.: SNOPT, A large-scale smooth optimization problems having linear or nonlinear objectives and constraints, http://www-neos.mcs.anl.gov/neos/solvers
9. Stone, J.R., Groesbeck, D.E., Zola Charles, L.: An improved prediction method for noise generated by conventional profil coaxial jets. National Aeronautics and Space Administration. Report NASA-TM-82712, AIAA-1991 (1981)
10. Wright, S.: Interior-point methods for optimal control of discrete-time systems. J. Optim. Theory Appls. 77, 161–187 (1993)
11. Wright, S.: Primal-dual interior-point methods. Society for Industrial and Applied Mathematics. SIAM, Philadelphia (1997)
12. Zaporozhets, O.I., Tokarev, V.I.: Aircraft Noise Modelling for Environmental Assessment Around Airports. Applied Acoustics 55(2), 99–127 (1998)
13. Zaporozhets, O.I., Tokarev, V.I.: Predicted Flight Procedures for Minimum Noise Impact. Applied Acoustics 55(2), 129–143 (1998)

Scalability Analysis of a Novel Integer Programming Model to Deal with Energy Consumption in Heterogeneous Wireless Sensor Networks

Alexei Aguiar, Plácido Rogério Pinheiro, André L.V. Coelho, Napoleão Nepomuceno, Álvaro Neto, and Ruddy P.P. Cunha

Universidade de Fortaleza, Mestrado em Informática Aplicada
Av. Washington Soares 1321, Sala J-30, Fortaleza, CE, Brasil, 60811-905
{placido,acoelho}@unifor.br,
{alexeiaguiar,napoleao,netosobreira,ruddypaz}@edu.unifor.br

Abstract. This paper presents a scalability analysis over a novel integer programming model devoted to optimize power consumption efficiency in heterogeneous wireless sensor networks. This model is based upon a schedule of sensor allocation plans in multiple time intervals subject to coverage and connectivity constraints. By turning off a specific set of redundant sensors in each time interval, it is possible to reduce the total energy consumption in the network and, at the same time, avoid partitioning the whole network by losing some strategic sensors too prematurely. Since the network is heterogeneous, sensors can sense different phenomena from different demand points, with different sample rates. As the problem instances grows the time spent to the execution turns impractcable.

1 Introduction

Wireless sensor networks (WSNs) have been primarily used in the monitoring of several physical phenomena, such as temperature, barometric pressure, humidity, ambient light, sound volume, solar radiation, and precipitation, and therefore have been deployed in different areas of application/research, like agriculture, climate study, biology, and security.

The simple deployment of the approach proposed by Nakamura et al. [6], while sensing different phenomena through the same WSN, can lead to inefficiency in terms of energy expenditure. With this perspective in mind, in this work, we provide an extension to the model devised by Nakamura et al. [6], namely, to consider different coverage radius and sampling rates for different phenomena. We argue that the incorporation of such aspects into the model can have a significant impact on the network lifetime mainly when the spatio-temporal properties of the phenomena under observation vary a lot. The introduction of this new dimension into the model brings about novel issues to be dealt with. The critical issue relates to the

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 11–20, 2008.

concurrent routing of data related to different phenomena, as these data should be relayed to different sinks.

The rest of the paper is organized as follows. Section 2 presents the WSN, how do they work, the components of a sensor, the problems that can occur in a WSN and complementary knowledge to optimize the Network. Section 3 presents the novel integer linear programming model for the minimization of energy expenditure in WSNs regarding the heterogeneity aspects of the sensed phenomena mentioned above. Section 4 presents initial results achieved by simulation while Section 5 provides a qualitative discussion of such results. Finally, Section 6 concludes the paper and comments on future work.

2 The Wireless Sensor Network

A Wireless Sensor network typically consist of a large number of small, low-power, and limited-bandwidth computational devices, named sensor nodes. These nodes can frequently interact with each other, in a wireless manner, in order to relay the sensed data towards one or more processing machines (a.k.a. sinks) residing outside the network. For such a purpose, special devices, called gateways, are also employed, in order to interface the WSN with a wired, transport network. To avoid bottleneck and reliability problems, it is pertinent to make one or more of these gateways available in the same network setting, a strategy that can also reduce the length of the traffic routes across the network and consequently lower the overall energy consumption. A typical sensor node is composed of four modules, namely the processing module, the battery, the transceiver module and the sensor module [4]. Besides the packet building processing, a dynamic routing algorithm runs over the sensor nodes in order to discover and configure in runtime the "best" network topology in terms of number of retransmissions and waste of energy. Due to the limited resources available to the microprocessor, most devices make use of a small operating system that supplies basic functionalities to the application program. To supply the power necessary to the whole unit, there is a battery, whose lifetime duration depends on several aspects, among which, its storage capacity and the levels of electrical current employed in the device. The transceiver module, conversely, is a device that transmits and receives data using radio-frequency propagation as media, and typically involves two circuits, viz. the transmitter and the receiver. Due to the use of public-frequency bands, other devices in the neighborhood can cause interference during sensor communication. Likewise, the operation/interaction among other sensor nodes of the same network can cause this sort of interference. So, the lower is the number of active sensors in the network, the more reliable tends to be the radio-frequency communication among these sensors. The last component, the sensor module, is responsible to gauge the phenomena of interest; the ability of concurrently collecting data pertaining to different phenomena is a property already available in some models of sensor nodes.

For each application scenario, the network designer has to consider the rate of variation for each sensed phenomenon in order to choose the best sampling rate of each sensor device. Such decision is very important to be pursued with precision as it surely has a great impact on the amount of data to be sensed and delivered, and, consequently, on the levels of energy consumed prematurely by the sensor nodes. This is the temporal aspect to be considered in the network design.

Another aspect to be considered is the spatial one. Megerian et al. [5] define coverage as a measure of the ability to detect objects within a sensor field. The lower the variation of the physical variable being measured across the area, the shorter has to be the radius of coverage for each sensor while measuring the phenomenon. This will have an influence in the number of active sensors to be employed to cover all demand points related to the given phenomenon. The fact is: the more sensors are active in a given moment, the bigger is the overall energy consumed across the net. WSNs are usually deployed in hostile environments, with many restrictions of access. In such cases, the network would be very unreliable and unstable if the minimum number of sensor nodes was effectively used to cover the whole area of observation. If some sensor node fails to operate, its area of coverage would be out of monitoring, preventing the correlation of data coming from this area with others coming from other areas.

Another worst-case scenario occurs when we have sensor nodes as network bottlenecks, being responsible for routing all data coming from the sensor nodes in the neighborhood. In this case, a failure in such nodes could jeopardize the whole network deployment. To avoid these problems and make a robust design of the WSN, extra sensor nodes are usually employed in order to introduce some sort of redundancy. By this means, the routing topology needs to be dynamic and adaptive: When a sensor node that is routing data from other nodes fails, the routing algorithm discovers all its neighbor nodes and then the network reconfigures its own topology dynamically. One problem with this approach is that it entails unnecessary energy consumption. This is because the coverage areas of the redundant sensor nodes overlap too much, giving birth to redundant data. And these redundant data bring about extra energy consumption in retransmission nodes. The radio-frequency interference is also stronger, which can cause unnecessary retransmissions of data, increasing the levels of energy expenditure. Megerian and Potkonjak [1] present many integer linear programming models to maximize energy consumption but not consider the dynamic time scheduling.

The solution proposed by Nakamura et al. [6] is to create different schedules, each one associated with a given time interval, that activate only the minimum set of sensor nodes necessary to satisfy the coverage and connectivity constraints. The employment of different schedules prevents the premature starvation from some of the nodes, bringing about a more homogeneous level of consumption of battery across the whole network. This is because the alternation of active

nodes among the schedules is often an outcome of the model, as it optimizes the energy consumption of the whole network taking into account all time intervals and coverage and connectivity constraints. It is well-known that the sensing of different phenomena does not follow the same spatio-temporal profile. For instance, the temporal and spatial variations of temperature measurements in a given area can be very different from those related to humidity. Working with only one radius of coverage for all sensed phenomena entails that this radius be the smallest one. Likewise, choosing only one sampling rate for all sensed phenomena implies that this rate can keep up well with the phenomenon that varies faster.

3 Model for Optimizing the Energy Consumption

In order to properly model the heterogeneous WSN setting, some previous remarks are necessary:

1. A demand point is a geographical point in the region of monitoring where one or more phenomena are sensed. The distribution of such points across the area of monitoring can be regular, like a grid, but can also be random in nature. The density of such points varies according to the spatial variation of the phenomenon under observation. At least one sensor must be active in a given moment to sense each demand point. Such constraint is implemented in the model;
2. Usually, the sensors are associated with coverage areas that cannot be estimated with accuracy. To simplify the modeling, we assume plain areas without obstacles. Moreover, we assume a circular coverage area with a radius determined by the spatial variation of the sensed phenomenon. Within this area, it is assumed that all demand points can be sensed. The radio-frequency propagation in real WSNs is also irregular in nature. In the same way, we can assume a circular communication area. The radius of this circle is the maximum distance at which two sensor nodes can interact;
3. A route is a path from one sensor node to a sink possibly passing through one or more other sensor nodes by retransmission. Gateways are regarded as special sensor nodes whose role is only to interface with the sinks. Each phenomenon sensed in a node has its data associated with a route leading to a given sink, which is independent from the routes followed by the data related to other phenomena sensed in the same sensor node;
4. The energy consumption is actually the electric current drawn by a circuit in a given time period.

In what follows, the elements of the novel integer linear programming model are introduced in a step-by-step manner.

S Set of sensors
D Set of demand points
M Set of sinks
G Set of phenomena (temperature, humidity, barometric pressure, etc.). Each phenomenon has its own spatio-temporal properties. The associated sampling rate has impact on data traffic, while the associated radius of coverage has impact on the number of active sensors
t Number scheduling periods
A^d Set of arcs that link sensors to demand points for phenomena
A^s Set of arcs that interconnects sensors
A^m Set of arcs that link sensors and sinks
$E^d(A)$ Set of incident arcs for demand point $d \in D$ which belong to A
$E^s(A)$ Set of incident arcs for sensor $s \in S$ which belong to A
$S^s(A)$ Set of output arcs leaving sensor $s \in S$ which belong to A
EB_i Cumulated battery energy for sensor $i \in S$
EA_i Energy dissipated while activating sensor $i \in S$
EM_i Energy dissipated while sensor $i \in S$ is activated (effectively sensing)
ET_{ij}^g Energy dissipated when transmitting data from sensor i to sensor j with respect to phenomenon g. Such values can be different for each arc ij if a sensor can have its transmitter power adjusted based on the distance to the destination sensor. Each phenomenon has its own sampling rate, a parameter that impacts the total amount of data transmitted across the WSN and, consequently, the levels of energy waste
ER_I Energy expended in the reception of data for sensor $i \in S$
EH_J^G Penalty applied when a demand point $j \in D$ for phenomenon g is not covered by any sensor
EG_i^g Penalty applied when sensor $i \in S$ is activated to unnecessarily sense the phenomenon g
x_{ij}^{tg} if sensor i covers demand point j in period t for phenomenon g
z_{lij}^{tg} if arc ij belongs to the route from sensor l to a sink in period t for phenomenon g
w_l^t if sensor i was activated in period t for at least one phenomenon
r_i^{tg} if sensor i was activated in period t for phenomenon g
y_i^t if sensor i is activated in period t
h_j^{tg} if demand point j for phenomenon g is not covered by any sensor in period t
e_i Energy consumed by sensor i considering all time periods

The objective function (1) minimizes the total energy consumption through all time periods. The second term penalizes the existence some not-covered demand points, but the solution continues feasible. It penalizes unnecessary activation for phenomenon too.

$$\min \sum_{i \in S} e_i + \sum_{t \in T} \sum_{g \in G} \left(\sum_{j \in D} EH_j^t h_j^{tg} + \sum_{i \in S} EG_i^{tg} r_i^{tg} \right) \tag{1}$$

These are the constraints adopted:

$$\sum_{ij \in E_j^d(A_g^d)} x_{ij}^{tg} + h_j^{tg} \geq 1, \forall j \in D, \forall t \in T, \forall g \in G \tag{2}$$

Constraint (2) enforces the activation of at least one sensor node i to cover the demand point j associated with phenomenon g in period t. Otherwise, the penalty variable h is set to one. This last condition will occur only in those cases when no sensor node can cover the demand point.

$$x_{ij}^{tg} \leq r_i^{tg}, \forall i \in S, \forall ij \in A_g^d, \forall t \in T, \forall g \in G \tag{3}$$

Constraint (3) turns on variable r (which means that a sensor node is actively sensing phenomenon g in period t) if its associated sensor node is indeed allocated to cover any demand point associated with g.

$$r_i^{tg} \leq y_i^t, \forall i \in S, \forall t \in T, \forall g \in G \tag{4}$$

Constraint (4) reads that sensor node i is fully active (parameter y), if it is active for at least one phenomenon of observation.

$$\sum_{ij \in E_j^s(A^s)} z_{lij}^{tg} - \sum_{jk \in S_j^s(A^s \cup A^m)} z_{ljk}^{tg} = 0, \forall j \in (S \cup M - l), \forall l \in S, \forall t \in T, \forall g \in G \tag{5}$$

Constraint (5) relates to the connectivity issue using the flow conservation principle. This constraint enforces that an outgoing route exists from sensor node j to sensor node k if there is already an incoming route from sensor node i to sensor node j.

$$- \sum_{jk \in S_j^s(A^s \cup A^m)} z_{ljk}^{tg} = -r_l^{tg}, j = l, \forall l \in S, \forall t \in T, \forall g \in G \tag{6}$$

Constraint (6) enforces that a route is created for phenomenon g if a sensor node is already active for that phenomenon.

$$z_{lij}^{tg} \leq y_i^t, \forall i \in S, \forall l \in (S - j), \forall ij \in (A^s \cup A^M), \forall t \in T, \forall g \in G \tag{7}$$

In Constraint (7), if there is an outgoing route passing through sensor node i, then this sensor node has to be necessarily active.

$$z_{lij}^{tg} \leq y_i^t, \forall j \in S, \forall l \in (S-j), \forall ij \in (A^s \cup A^M), \forall t \in T, \forall g \in G \tag{8}$$

In the same way, with Constraint (8), if there is an incoming route passing through sensor i, then this sensor has to be active.

$$\sum_{t \in T} \sum_{g \in G} (EM_i y_i^t + EA_i w_i^t + \sum_{l \in (S-i)} \sum_{ki \in E_i^s (A^s \cup A^m)} ER_i z_{lki}^{tg} + \sum_{l \in S} \sum_{ij \in S_i^s (A^s \cup A^m)} ET_i^g j z_l^t ij) \leq e_i, \forall i \in S \tag{9}$$

The total energy consumed by a sensor node is the sum of the parcels given in Constraint (9).

$$0 \leq e_i \leq EB_i, \forall i \in S \tag{10}$$

Constraint (10) enforces that each sensor node should consume at most the energy capacity limit of its battery.

$$w_i^0 - y_i^0 \geq 0, \forall i \in S \tag{11}$$

Constraint (11) determines when the sensor node should start to sense (parameter w). If a sensor is active in the first period, its corresponding w should be set to 1.

$$w_i^t - y_i^t + y_i^{t-1} \geq 0, \forall i \in S, \forall t \in T, t > 0 \tag{12}$$

In Constraint (12), the past and current activation states of a sensor node are compared. If the sensor node was active from period $t-1$ to period t, then w is set to 1.

$$x, y, z, w, h \in \{0, 1\}, e \in \mathbb{R}. \tag{13}$$

4 Computational Results

In order to assess the potentialities of the novel optimization model, we have devised the simulation scenario that is described in the sequence. First of all, we have considered only two phenomena of interest to be concurrently sensed by the same WSN. Besides, only four time intervals were taken into consideration to alleviate the computational burden, although the reader should be aware that the real benefits of our extended model appear (that is, the savings in terms of energy expenditure would be more significant) when one has to deal with larger numbers of time intervals.

A regular grid of demand points was considered: There were 100 demand points in a square area of 10 per 10 meters, with one demand point per square meter. Each demand point can be assigned to either or both phenomena, but

the overall coverage of each phenomenon is totally independent from each other regarding a demand point alone. In the same vein, sixteen sensor nodes were placed in a regular 4 × 4 grid. All nodes have the same processing/sensing capabilities with the possibility to sense concurrently the two phenomena. The coverage radius for the first phenomenon was set as 8.8 meters in length while the length of the coverage radius for the second phenomenon was 16 meters. The sampling rate for the first and second phenomena was set as two samples per minute and one sample per minute, respectively. The length of the radius of communication between two neighbor sensors was 11 meters in size. Only one sink was placed at the middle of the regular grid. All elements of this scenario (demand points, sensors, and sink) were generated with its associated geographic coordinates. The matrix was filled with ones in those cases where the distance from the sensor and the demand point was less than or equal to the coverage radius for each phenomenon, and with zeros otherwise. Similarly, the matrices and were filled with ones in those positions where the distance between the sensor nodes or from a sensor node to the sink was less than or equal to the communication radius, and with zeros otherwise. The energy constants were calculated having as basis the values announced at a spreadsheet from a sensor node manufacturer [2]. The energy values for transmission and reception were calculated having as basis the amount of sensed data and the bit rate adopted in the devices. The penalty constant was assigned to a high value to enforce that the model covers all demand points of interest.

In order to establish a comparison, in terms of problem difficulty (variables and constraints) and energy savings (objective function values), between the heterogeneous WSN setting and its homogeneous counterpart, we have also conducted some simulations with our model considering two phenomena with the same characteristics, namely coverage radius of 8.8 meters and sampling rate of two samples per minute.

Table 4 shows the simulation results achieved by playing with the CPLEX platform [3] with OPL Development Studio 4.2 and Cplex 10.0. The tests were executed in Pentium D 3 GHz 512 MB machines with Windows XP Professional. In this table, in the calculus of the "real objective" value we ignore the penalties and sum up only the variables. Figures 1 and 2 provide snapshots of the scheduled plans generated for the first and second phenomena regarding the four time intervals considered.

In a manner as to have a better feeling of the impact of the data routing process on the energy expenditure of the WSN nodes, we have set up a second scenario with a larger area, where the length of the coverage and communication radii become smaller. By this means, there are few communication options to each sensor, and routes must be established in order to convey data to the sinks. In this new scenario, there are four sinks in the corners of the square area and our aim is to assess how many sensor nodes the model recruits to operate as routers of the traffic towards the sinks. Figure 3 the routes generated to this scenario by our model.

Table 1. Simulation results comparing homogeneous and heterogeneous WSN settings

Model	Homogen.	Heterogen.	Homogen.	Heterogen.	Homogen.	Heterogen.
Phenomenom	1	2	1	2	1	2
Time intervals	4	4	5	5	6	6
Demand pts.	100	100	100	100	100	100
Sensor nodes	16	16	16	16	16	16
Sinks	1	1	1	1	1	1
Variables	24,416	48,688	30,517	60,857	36,617	73,025
Constraints	39,248	78,384	49,048	97,968	58,848	117,552
Density	9.47e-05	4.89e-05	7.58e-05	3.92e-05	6.32e-05	3.26e-05
Time (h:m:s)	00:00:27	00:01:57	00:02:13	15:28:02	00:05:24	23:30:57
Objective	16,155.120	15,053.768	21,409.072	19,643.030	26,232.680	24,017.120
Real objective	16,155.120	12,653.768	21,409.072	16,543.200	26,232.680	20,825.570
Reduction	0%	21.67%	0%	15,78%	0%	20.61%

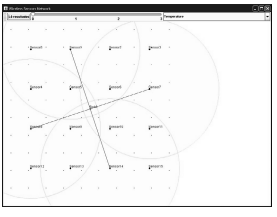


Fig. 1. First phenomenon intervals 1-3

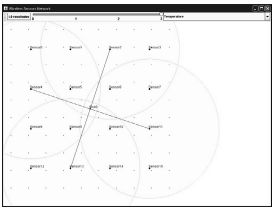


Fig. 2. First phenomenon interval 4

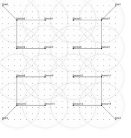


Fig. 3. Routes generated by the model

5 Discussion

Evaluating the results presented in the preceding section, one can notice that the deployment of a homogeneous WSN setting has entailed an overall energy consumption of 16,155.12 mAh, which is much higher than the 12,653.768 mAh level achieved with the heterogeneous model. That is, by employing the extended optimization model, 21.67% of energy savings could be obtained. As mentioned above, this gain is mainly due to the unnecessary generation and transmission of data when the spatio-temporal properties of the sensed phenomena are treated as the same. These simulation results corroborate with our perspective that, the more different are the phenomena sensed by the same WSN, the higher tends to be the overall gain in making use of the heterogeneous approach. A drawback exhibited by the model during the simulation runs was that of shortage of scalability. We postulate that a much noticeable increase in the network lifetime could be achieved if we had increased the number of time intervals considered. However, when we have tried to exploit this expedient, the computational burden of the simulation has increased a lot. This is indeed a limitation of the approach that demands to be circumvented in future work.

6 Conclusion and Future Work

In this paper, a novel integer linear programming model devoted to optimize power consumption efficiency in heterogeneous wireless sensor networks is presented. This model is based upon a schedule of sensor allocation plans in multiple time intervals subject to coverage and connectivity constraints. By turning off a specific set of redundant sensors in each time interval, it is possible to reduce the total energy consumption in the network and, at the same time, avoid partitioning the whole network by losing some strategic sensors too prematurely. This sensor activity alternation in time intervals can yield a properly-adjusted balance in terms of energy expenditure among the sensors, overcoming some of the problems usually incurred with the deployment of static plans. The current work can be enhanced by making use of alternative optimization methodologies. It is possible to allow larger problem instances to be solved in much viable times and by adding new features to the model. We are currently investigating the deployment of novel methodologies hybridizing metaheuristics with exact methods, such as the one recently proposed by Nepomuceno et al. [8,7], which has achieved very good results in other classes of hard optimization problems. We feel that, by customizing the methodology to deal with the heterogeneous WSN configuration problem, can let us raise the complexity of the simulated problem instances and investigate new interesting aspects.

References

[1] Megerian, S., Potkonjak, M.: Lower power 0/1 coverage and scheduling techniques in sensor networks. Technical Reports Vol. 030001. University of California, Los Angeles (2003)
[2] Mote battery life calculator, http://www.xbow.com/Support/Sypport_pdf_files/PowerManagement.xls
[3] ILOG: ILOG CPLEX 9.0 User's Manual (2003)
[4] Loureiro, A., Ruiz, L., Mini, R., Nogueira, J.: Redes de sensores sem fio. Simpósio Brasileiro de Computação, Jornada de Atualização de Informática (2002)
[5] Megerian, S., Koushanfar, F., Qu, G., Veltri, G., Potkonjak, M.: Exposure in wireless sensor networks: Theory and practical solutions. Wireless Networks 8(5), 443–454 (2002)
[6] Nakamura, F.G., Quintão, F.P., Menezes, G.C., Mateus, G.R.: Planejamento dinâmico para controle de cobertura e conectividade em redes de sensores sem fio. In: Workshop de Comunicação sem Fio e Computação Móvel, vol. 1, pp. 182–191 (2004)
[7] Nepomuceno, N.V., Pinheiro, P.R., Coelho, A.L.V.: A Hybrid Optimization Framework for Cutting and Packing Problems: Case Study on Constrained 2D Non-guillotine Cutting. In: Cotta, C., Hemert., J. (eds.) Recent Advances in Evolutionary Computation for Combinatorial Optimization. Springer, Heidelberg (to appear)
[8] Nepomuceno, N., Pinheiro, P.R., Coelho, A.L.V.: Tackling the container loading problem: A hybrid approach based on integer linear programming and genetic algorithms. In: Cotta, C., van Hemert, J. (eds.) EvoCOP 2007. LNCS, vol. 4446, pp. 154–165. Springer, Heidelberg (2007)

Single Straddle Carrier Routing Problem in Port Container Terminals: Mathematical Model and Solving Approaches

Babacar Mbaye Ndiaye[1], Pham Dinh Tao[2], and Hoai An Le Thi[3]

[1] Laboratory of LMDAN, FASEG,
University of Cheikh Anta Diop, Dakar
BP 5683, Dakar-Fann Senegal
bmndiaye@ucad.sn
[2] Laboratory of Modelling, Optimization & Operations Research,
National Institute for Applied Sciences - Rouen, BP 08, Place
Emile Blondel 76131, Mont Saint Aignan Cedex, France
pham@insa-rouen.fr
[3] Laboratory of Theoretical and Applied Computer Science
UFR MIM, University of Paul Verlaine, Metz,
Ile du Saulcy, 57045 Metz, France
lethi@sciences.univ-metz.fr

Abstract. It is discussed how to route straddle carriers during the loading operation of export containers. The straddle carrier (SC) travel distance will be much longer if two consecutive containers must be collected far from one another instead of from consecutive yard-bays. So, finding the minimum straddle carrier travel distance will guarantee port efficiency and allow cost savings. Our objective is to minimize this total travel distance of a straddle carrier. A SC performs a so called partial tour to pick-up containers of a same group, according to the work schedule. This problem is characterized as problem with binary variables, which are hard to solve optimally.

In this paper we reformulate the problem, thanks to exact penalty techniques in DC Programming, as a polyhedral DC Program. A combination of the local algorithm DCA and global optimization approach such as Cutting plane techniques is proposed. The performance of the algorithm is tested on a set of data and the computational results are presented.

Keywords: Vehicle Routing, Container port, Straddle carrier, Nonconvex optimization-Global optimization, DC (Difference of Convex Functions) Programming, DCA (DC Algorithm), Cutting plane techniques.

1 Introduction

Port operations concerning containers essentially comprehend loading, stocking and transferring which have a direct impact on port service and performance

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 21–31, 2008.

of the entire terminal configuration. This port scenario involves several computational problems, such as allocation of containers in port yards and also in ships, as well as single straddle carrier routing problem (SSCRP) and manipulation of containers. Loading export containers on a ship requires three types of equipments: straddle carriers, yard trucks and quay cranes.

The straddle carrier has to move containers from where they are stored within a container terminal and to deliver them to a yard truck. The yard truck, a combination of yard tractor and yard trailer, has to transport containers received from a straddle carrier to the marshalling area. The quay crane (QC) has to pick up containers from the marshalling area and to place them inside container ships. It is a static equipment. A container terminal yard is subdivided into blocks of yard-bays which contain containers arranged in rows. A yard-map shows the distances between blocks and between consecutive yard-bays.

Each straddle carrier, the only equipment allowed to enter yard-bays, is assigned to fulfil a quay crane loading sequence. This sequence defines a work schedule which determines the exact order in which containers must be handled and delivered by a straddle carrier (and consequently by a yard truck) to a quay crane.

For the study of the problem of straddle carrier routing problem and the global movement management in port container terminals, see [1,2,3,4].

In port container terminal, Kozan and Preston [1] proposed an analytical scheduling model, and a solution based on genetic algorithm, in order to minimize handling for loading export containers into ships. Kim and Kim [3] formulated the port routing problem for export containers during the loading process using Integer Programming. A Dynamic Programming solution was proposed to minimize the total travel distance of a single straddle carrier vehicle, used for container transportation between storage and marshalling areas. In [2] they proposed a Beam Search procedure for resolving the same routing problem. An evaluation of algorithm performance was discussed based on numerical experimentation and comparison between Beam Search and Genetic Algorithm results.

In this paper, we developed the new approaches of DC Programming and DCA for numerical processing of this class, very significant and difficult, of non convex problems. Our algorithm converges to a local solution after many finite iterations and it consists of solving a linear program, at each iteration. Moreover, although our DCA is a continuous approach that works on a continuous domain, it provides an integer solution.

DC Algorithm (DCA), based on local optimality conditions and the duality in DC programming, have been introduced by Pham Dinh in 1986 [5] as an extension of the sub gradient algorithms to DC programming. Important improvements and developments for DCA from both theorical and computational aspects have been completed since 1994 by Le Thi and Pham Dinh ([11,12,13,14]; and references therein).

In section 2 and 3, a formulation and an algorithm are suggested. In section 4, computational experimentations are presented. Finally, summary and conclusions are provided in section 5.

2 The Problem Formulation

The formulation of routing straddle carriers for the loading operation of containers in automated container terminal has been presented in [2].

Notations

m : number of partial tours for a SC complete tour,
n : number of yard-bays, **l** : number of container groups,
t : partial-tour number, t=0,1,...,m,m+1; where t=0 and t=m+1 at source and terminal vertices in the network representation,
B : set of indexes of yard-bays = {1,2, ... n},
G : set of indexes of container groups = {1,2, ... l},
S(h) : set of indexes of partial tours corresponding to container group h,
B(h) : set of yard-bay numbers which contain containers of group h,
$\mathbf{c_{hj}}$: initial number of containers of group h stacked at yard-bay j,
$\mathbf{r_t}$: number of containers to pick up during partial tour t,
$\mathbf{g_t}$: container group number to be picked up during partial tour t,
$\mathbf{d_{ij}}$: travel distance between yard-bays i and j,
$\mathbf{B_{g_0}}$ = {S} source, $[\mathbf{B_{g_{m+1}}}]$= {T} sink, and **M** a very large number.

– variable $\mathbf{y} = (\mathbf{Y^t_{ij}})$,

$$Y_{ij}^t = \begin{cases} 1 \text{ if SC moves from yard-bay i to j after completing partial-tour t} \\ 0 \text{ otherwise.} \end{cases}$$

– variable $\mathbf{z} = (\mathbf{Z^t_{ij}})$,

$$Z_{ij}^t = \begin{cases} 1 \text{ if SC moves from yard-bay i to j during a partial-tour t} \\ 0 \text{ otherwise.} \end{cases}$$

– variable $\mathbf{x} = (\mathbf{X^t_j})$,

X_j^t = number of containers picked-up at yard-bay j during partial tour t

A tour t is defined as a visiting sequence of yard-bays by a SC in order to pick up all the specified containers in the corresponding work schedule. A partial-tour of a SC is the visiting sequence of yard-bays during which a SC picks up all the containers for a cluster of cells in a ship.

• The problem, denoted by ($\mathcal{P}$), can be formulated as :
The objective function that minimizes the total distances travelled between partial-tours and within a partial-tour.

$$\min_{X_j^t, Y_{ij}^t, Z_{ij}^t} \sum_{t=0}^{m} \sum_{i \in B(g_t), j \in B(g_{t+1})} d_{ij} Y_{ij}^t + \sum_{t=1}^{m} \sum_{(i,j) \in B(g_t)} d_{ij} Z_{ij}^t$$

s.t.

(i) represents the gain of flows at the source node

$$\sum_{i \in B(g_t)} Y_{Sj}^0 = 1 \tag{1}$$

(*ii*) represents the gain of flows at the terminal node

$$-\sum_{j\in B(g_m)} Y_{jT}^m = -1 \tag{2}$$

(*iii*) represents the flow conservation at the other nodes

$$\sum_{j\in B(g_{t-1}),k\in B(g_t)} (Y_{ji}^{t-1} + Z_{ki}^t) - \sum_{j\in B(g_{t+1}),k\in B(g_t)} (Y_{ij}^t + Z_{ik}^t) = 0 \tag{3}$$

$$\forall i \in B(g_t), \quad \forall t = 1, 2 \ldots, m$$

(*iv*) prevents the looping of sub-tours. An isolated cycle may exist in the final solution that is not connected to the path from the source node to the terminal node.

$$\sum_{(i,j)\in B(g_t)} Z_{ij}^t < |N| - 1 \qquad \forall N \subseteq B(g_t), \quad \forall t = 1, 2 \ldots, m \tag{4}$$

(*v*) implies that only when a SC visits a yard-bay can it pick up containers at the yard-bay.

$$X_j^t \leq M(\sum_{k\in B(g_t)} Z_{kj}^t + \sum_{i\in B(g_{t-1})} Y_{ij}^{t-1}) \qquad \forall j \in B(g_t), \quad \forall t = 1, 2 \ldots, m \tag{5}$$

(*vi*) implies that the number of containers picked up in a partial-tour should be equal to the number of containers requested by a work schedule.

$$\sum_{j\in B(g_t)} X_j^t = r_t \qquad \forall t = 1, 2 \ldots, m \tag{6}$$

(*vii*) means that the total number of containers picked up during the whole tour should be equal to the initial number of containers at each bay for each specific container group

$$\sum_{t\in S(h)} X_j^t = c_{hj} \qquad j \in B(g_t), \quad \forall h = 1, 2 \ldots, l \tag{7}$$

(*viii*) represent the domains of the variables.

$$X_j^t \in \mathbb{N} \qquad \forall j \in B(g_t), \quad \forall t = 1, 2, \ldots, m \tag{8}$$

$$Y_{ij}^t \in \{0, 1\} \qquad \forall i \in B(g_t), \quad \forall j \in B(g_{t+1}), \quad \forall t = 0, 1, \ldots, m \tag{9}$$

$$Z_{ij}^t \in \{0, 1\} \qquad \forall i, j \in B(g_t), \quad \forall t = 1, 2 \ldots, m \tag{10}$$

3 A Combined DCA and New Cutting Plane Techniques

In this section, we propose a global method based on a new cutting planes with an original and robust local approach namely DCA. Contrary to the classical approaches [8,16]; we solve an equivalent problem with continue variables, thanks to exact penalty techniques in DC Programming. Our cutting plane is obtained from a local minimum of the penalty function in the relaxed domain of $(\mathcal{P})$. It's a new method developed in [15], where comparison results therein, in non-convex real problems have shown the efficiency of this algorithm.

DC reformulation of the problem $(\mathcal{P})$
Using the well known results concerning the exact penalty, we will formulate the problem $(\mathcal{P})$ in the form of a DC program. Let:

• $C = (C_1, C_2)$, $Y = (Y_{ij}^t)$, $Z = (Z_{ij}^t)$ $V = (Y, Z) \in \mathbb{R}^{n=n_1+n_2}$, $X = (X_j^t)$, and $U = (V, X) \in \mathbb{R}^{n+p}$. C_1 and C_2 represent the cost of the first and the second member in the objective function, respectively.
• K the set of feasible points U=(V,X) determined by the system of the constraints $\{(1), \ldots, (8)\}$, $S = \{U = (V, X) \in K : 0 \leq V \leq 1\}$, is nonempty, bounded polyhedral convex set in $\mathbb{R}^{n+p}$.
• $p(V, X) = p(U) = \sum_{i=1}^{n} \min(V_i, 1 - V_i)$. It is clear that p is concave and finite on S, and $p(U) \geq 0$ for all $U \in S$.

The problem $(\mathcal{P})$ can be expressed in the form:

$$\alpha = \min\{C^T V : U = (V, X) \in K, V \in \{0, 1\}^n\} \tag{11}$$

Problem (11) can be rewritten as follows:

$$\alpha = \min\{C^T V : U \in S, p(U) \leq 0\} \tag{12}$$

From the Theorem of the exact penalty (Theorem 1, [6]), we get, for a sufficiently large number t $(t \geq t_0)$, the equivalent concave minimization problem to (11):

$$\alpha(t) = \min\{C^T V + tp(U) : U \in S\} \tag{13}$$

$$= \min\{g(U) - h(U) = f(U) : U \in \mathbb{R}^{n+p}\} \tag{14}$$

with $g(U) = \chi_S(U)$ and $h(U) = \langle -C, U\rangle - t \sum_{i=1}^{n} \min(V_i, 1 - V_i)$
where: χ_S is the indicator function of S, $S = \{U \in K : 0 \leq V \leq 1\}$.

It is clear that g and h are two convex functions, and so problem (13) is a DC program, in the form expressed in (14).

Briefly, DCA is a descent method and converges to a critical point of $g - h$. If either g or h is polyhedral convex, then (14) is called a polyhedral DC program for which DCA has a finite convergence [7,9]. That is the case for DCA applied

to (13). Convergence properties of DCA and its theoretical basis can be found in [7,9,12]. Our algorithm converges to a critical point (to local solution in almost cases) after a many finitely iterations and its consists of solving a linear program at each iteration.

3.1 Construction of a Cutting Plane from a Local Solution of the Penalty Function

Let $u^* = (v^*, x^*)$ the solution of DCA applied to (14). We have the following two cases: first, u^* is a feasible solution of the original problem $(\mathcal{P})$, i.e., $v^* \in \{0,1\}^n$. Second, u^* is not feasible, i.e., at least there exist an index j_0 such that $v^*_{j_0}$ is a rational number. The case (i) will be discuss in the next section. We just take account, in this section, of the case (ii).

• We consider the linear programming problem with mixed 0-1 variables:

$$(\mathcal{P}) \qquad \min\{C_1^T y + C_2^T z : Av + Bx \leq b, v \in \{0,1\}^n, x \in \mathbb{R}^p\}.$$

Now, let: $K := \{u = (v,x) \in \{0,1\}^n \times \mathbb{R}^p : Av + Bx \leq b\}$ and $S := \{u = (v,x) \in [0,1]^n \times \mathbb{R}^p : Av + Bx \leq b\}$. For all $u^* \in K$, we denote by:

$$I := \{1,\ldots,n\}; \quad J_0(u^*) := \{j \in \{1,\ldots,n\} : v_j \leq 1/2\}$$

$$J_1(u^*) := \{1,\ldots,n\} \setminus J_0(u^*), \quad \textit{and} \quad l_{u^*}(v) := \sum_{j \in J_0(u^*)} v_j + \sum_{j \in J_1(u^*)} (1 - v_j).$$

Let u^* be an infeasible local solution of the function $p(u) := \sum_{j=1}^{n} \min\{v_j, 1 - v_j\}$ on K, (i.e., u^* is a local solution of p on K and $u^* \notin S$).
Let:

$$\alpha = \min\{p(u) \mid u \in K\} \tag{15}$$

and we consider the following hypothesis.

Hypothesis (H): Let $u^0 \in \mathbb{R}^{n+p}$.
We suppose that we have an algorithm $\mathcal{A}$ that allows us to find a minimum $u^\star$ of the function p such that:

$$p(u^*) \leq p(u^0).$$

Theorem 1. *Let u^* be an infeasible local minimum of the function p on K. Suppose that $u^*_j \neq 1/2$ for all j. We have:*
(i) the following inequality is valid for $(\mathcal{P})$: $l_{u^*}(u) \geq l_{u^*}(u^*)$
(ii) if the value $p(u^*)$ is not integer then the inequality: $l_{u^*}(u) \geq \lfloor p(u^*) \rfloor + 1$ is a cutting plane which cuts off u^* from S.

Proof. For the proof of the theorem, see [15].

In the case where u^* is an infeasible local minimum of the problem (15) with the integer value $p(u^*)$: by applying the **Procedure P** (see [15,17]), we can obtain either a feasible solution or a cutting plane that cuts off u^* from S.
Procedure P stops after a finite number of iterations.

Remark 1. The cutting plane obtained, either thanks to the Theorem 1, or thanks to the Procedure P, which cuts off the minimum v^* can be written in the form:

$$l_{u^*}(v) = \sum_{i \in I_0(u^*)} v_i + \sum_{i \in I_1(u^*)} (1 - v_i) \geq \eta \tag{16}$$

where $\eta > l_{u^*}(v^*)$. Moreover, we can suppose that η is integer, otherwise replace η by $\lfloor \eta \rfloor + 1$.

3.2 Updating of the Upper Bound and the Best Known Solution - Separation of a Feasible Solution

We suppose that: at step k, $u^k = (v^k, x^k)$ is the best feasible solution known and γ_k is the upper bound ($\gamma_k = C_1^T y^k + C_2^T z^k$). In our approach, a feasible solution is often found after applying the Procedure P.

By convention, at the step k, if we know any feasible solution, then set $u^k = \emptyset$ and $\gamma_k = +\infty$. When a feasible solution is found, we update the best feasible solution and the upper bound. Suppose that u^k and γ_k are the best feasible solution and the upper bound, respectively, at the step k, $(k \geq 1)$.

If at the step $k+1$ we found the feasible solution u^* such that $C_1^T y^* + C_2^T z^* < C_1^T y^k + C_2^T z^k$, then we put $u^{k+1} = u^*$ and $\gamma_{k+1} = C_1^T y^* + C_2^T z^*$. Otherwise, we put $u^{k+1} = u^k$ and $\gamma_{k+1} = \gamma_k$.

Taking into account of the update of the best feasible solution and the upper bound, we separate this solution from the feasible set, to eliminate the solutions and the local minima already found by DCA.

The separation of a feasible solution w^k is carried out as follows:
We add to the problem $(\mathcal{P})$, a constraint (the separation constraint) in the form $h(u) \geq \zeta$ such as it eliminates only the point w^k of S, i.e.:

$$h(w) \geq \zeta \quad \forall w \in S \setminus \{w^k\} \qquad and \qquad h(w^k) < \zeta. \tag{17}$$

In our approach, this constraint is selected as follows:

$$h(w) \equiv h(v) := \sum_{j : v_j^k = 0} v_j + \sum_{j : v_j^k = 1} (1 - v_j) \geq 1 \tag{18}$$

It is easy to check that (18) satisfies the condition (17).

The DCA_CUT to solve the problem $(\mathcal{P})$ can be described by the next subsection.

3.3 Description of the Algorithm DCA_CUT

Algorithm 1. [DCA_CUT to solve the problem $(\mathcal{P})$**]**

Step 0. *(Initial)*
- $K_0 = K$; Let $\beta_0 \leftarrow -\infty$ (lower bound), $\gamma_0 \leftarrow +\infty$ (upper bound);

Step 1. *(Solving the linear problem)*
- Solve the linear problem $LB_k := \min\{C_1^T y + C_2^T z : (v, x) \in K_k\}$ to obtain the lower bound β_k and the optimal solution of existing relaxation u_{LP}^k;
- If $u_{LP}^k \in S$ then u_{LP}^k solve $(\mathcal{P})$; STOP.

Step 2. *(Apply DCA)*
- Apply DCA to the problem $\min\{C_1^T y + C_2^T z + tp(v) : (v, x) \in K_k\}$ to obtain its solution u_{DCA}^k;
- Call the **Procedure P**, if necessary, to obtain either a feasible solution or a cutting plane;
- If a feasible solution is obtained (denote by $w^k = u^k$), go to the Step 3; Otherwise go to the Step 4;

Step 3. *(Update the upper bound and the best known solution and separate* w^k*)*
- If $C_1^T y^k + C_2^T z^k < \gamma_k$, put $w_{Opt} \leftarrow w^k$; $\gamma_k \leftarrow C_1^T y^k + C_2^T z^k$;
- Separate w^k from S by addition of the inequality 18 to the constraints;

Step 4. *(Addition of the cutting plane)*
- Add the cutting plane obtained to the constraints;
- Put $k \leftarrow k + 1$; Go to the Step 1;

The convergence of Algorithm 1 is expressed through the following theorem:

Theorem 2. *Algorithm 1 converges to a global optimal solution after a finite number of iterations.*

For the proof of the theorem see [15,17].

4 Numerical Experiments

In order to determine a loading sequence, two documents are necessary. One is the yard-map that shows the distribution of containers of each container group in the yard. The other is the work schedule of each QC.

It is assumed that one SC is assigned to one QC. Since a containership is usually served by multiple QCs, containers in a yard are first allocated to an individual QC. Once all the containers are allocated to a specific QC, the carrier routing problem for each SC can be solved independently.

For more information on the datasets regarding technical parameters of the different technologies, see [2,3].

We denote by: $time(s)$ = total execution time in seconds, $iter$ = number of iterations, lb = lower bound, ub = upper bound, $gap = (ub - lb)/(1. + lb)$,

$nbcut$ = number of cuts, DCA_CUT = combination of DCA and Cutting Planes, $CPLEX$ 11.0 = CPLEX *Optimization* ILOG, version 11.0 [10].

The algorithm was implemented in C++, on the Toshiba computer: Intel(R) Pentium(R) 4 CPU 3.00GHz, 1.024Gb of RAM, under UNIX system.

To solve the linear programming, we used software CPLEX version 11.0.

To test the performance of the algorithm, we generated 15 realistic scenarios (based on a large number of parameters) supported in part by the Dakar Port, with 2 quay cranes, 5 container group quantity, 3 blocks and 8 yard-bays. In this case, there are 5423 variables with 4900 binary variables and 1124 constraints.

• The combined DCA_CUT is interesting, it is much better than the CPLEX 11.0 in terms of running time. The results show that DCA_CUT gives a good approximation of the optimal solution within a very short running time, compared to CPLEX 11.0. DCA intensely decreases the number of iterations and improves the upper bound. Our method usually provides a best upper bound after some iterations.

DCA_CUT is very fast and can then handle large-scale problems to improve the performance of the port terminal, for making possible to assist human experts to support real time decisions.

Table 1. DCA_CUT and CPLEX 11.0 for single straddle carrier routing problem

N_o	DCA_CUT						$CPLEX$ 11.0				
	$time(s)$	$iter$	lb	ub	gap	$nbcut$	$time(s)$	$iter$	lb	ub	gap
1	19.098	24	37.00	37.00	0.00	21	1292.013	813479	37.00	37.00	0.00
2	10.756	10	26.00	26.00	0.00	7	1777.153	524578	26.00	26.00	0.00
3	23.657	28	51.50	51.50	0.00	20	1085.613	453224	51.50	51.50	0.00
4	8.703	11	51.10	51.10	0.00	6	1074.447	1310164	51.10	51.10	0.00
5	15.540	19	51.40	51.40	0.00	14	1171.383	444561	51.40	51.40	0.00
6	22.001	27	51.05	51.05	0.00	21	1304.437	609681	51.05	51.05	0.00
7	19.110	23	60.00	60.05	0.00	20	1077.473	487331	60.00	60.00	0.00
8	17.563	19	66.00	66.00	0.00	16	1178.103	643191	66.00	66.00	0.00
9	24.141	25	28.60	28.65	0.00	19	1778.153	710017	28.60	28.60	0.00
10	13.319	14	47.60	47.60	0.00	9	2771.703	514019	47.60	47.60	0.00
11	25.533	27	52.20	52.20	0.00	21	1180.103	475211	52.20	52.20	0.00
12	19.328	21	52.00	52.00	0.00	17	2193.008	977123	52.00	52.00	0.00
13	20.711	26	52.50	52.50	0.00	21	617.713	1337561	52.50	52.50	0.00
14	17.052	17	58.05	58.05	0.00	14	2172.853	613442	58.05	58.05	0.00
15	19.991	24	61.40	61.40	0.00	21	1161.113	1300692	61.40	61.40	0.00

5 Summary and Conclusions

Port container terminal scenario involves several computational problems, such as allocation of containers in port yards and also in ships, as well as routing and manipulation of containers.

In this paper, we address the problem of the single straddle carrier routing problem in an export container terminal environment. We develop new

techniques to minimize the total travel distance of straddle carriers, which are used to transfer containers in a marshalling yard to yard trucks. Finding the minimum straddle carrier travel distance will guarantee port efficiency and allow cost savings.

We have developed a combination of DCA and new cutting plane technique to solve efficiently the mixed 0-1 linear program obtained. The numerical simulations on large scale datasets have shown the efficiency of our method. We show how the simulation technique of DCA_CUT can be applied successfully to practical decision support and employed as a decision support system for the terminal management.

References

1. Preston, P., Kozan, E.: Genetic algorithms to schedule container transfers at multimodal terminals. International Transactions in Operational Research (1999)
2. Kim, K.H., Kim, K.Y.: Routing straddle carriers for the loading operation of containers using a beam search algorithm. Computers and Industrial Engineering 36, 109–136 (1999)
3. Kim, K.Y., Kim, K.H.: A routing algorithm for a single straddle carrier to load export containers onto a containership. International Journal of Production Economics (1999)
4. Kim, K.H., Park, Y.M., Ryu, K.R.: Deriving decision rules to locate export containers in container yards. European Journal of Operational Research (2000)
5. Pham Dinh, T.: Algorithms for solving a class of nonconvex optimization problems.Methods of subgradients. Mathematics for Optimization. Math. Studies, 65–76 (1986)
6. Le Thi, H.A., Pham Dinh, T., Le Dung, M.: Exact penalty in dc programming. Vietnam Journal of Mathematics 27(2), 169–178 (1999)
7. Le Thi, H.A., Pham Dinh, T.: Convex analysis approach to d.c. programming: Theory, Algorithms and Applications. Acta Mathematica Vietnamica, 22(1), 289–355 (1997)
8. Marchand, H., Martin, A., Weismantel, R., Wolsey, L.: Cutting Planes in integer and mixed integer programming. Core discussion paper-9953, pp. 1–50 (1999)
9. Le Thi, H.A., Pham Dinh, T.: The DC Programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations Research 133, 23–46 (2005)
10. CPLEX Optimization ILOG, IncUsing the CPLEX^{R} Callable Library and CPLEX Barrier and Mixed Integer Solver Options, Version 11.0 (2007)
11. Le Thi, H.A., Pham Dinh, T.: DC optimization algorithms for solving the trust region subproblem. SIAM J. Optimisation 8, 476–505 (1998)
12. Le Thi, H.A., Pham Dinh, T.: A Continuous Approach for Globally Solving Linearly Constrained Quadratic Zero-One Programming Problems. Optimization 50, 93–120 (2001)
13. Le Thi, H.A., Pham Dinh, T.: Large Scale Molecular Optimization from exact distance matrices by a dc optimization approach. SIAM Journal on Optimization 14(1), 77–116 (2003)
14. Le Thi, H.A.: Contribution à l'optimisation non convexe et l'optimisation globale: Théorie, Algorithmes et Applications. Habilitation à diriger des recherches, Université de Rouen (Juin 1997)

15. Nguyen, V.V.: Contribution à l'optimisation non convexe et à l'optimisation globale en transport logistique. Thèse de doctorat, Institut National des Sciences Appliquées (INSA), Rouen (Juillet 2006)
16. Padberg, M.: Classical cuts for mixed-integer programming and branch-and-cut. Mathematical Methods of Operations Research (2), 173–203 (2001)
17. Ndiaye, B.M.: Simulation et Optimisation DC dans les réseaux de transport combiné. Code à usage industriel. Thèse de doctorat, Institut National des Sciences Appliquées (INSA), Rouen (Mai 2007)

Employing "Particle Swarm Optimization" and "Fuzzy Ranking Functions" for Direct Solution of EOQ Problem

Adil Baykasoğlu and Tolunay Göçken

University of Gaziantep, Department of Industrial Engineering, Gaziantep, Turkey
{baykasoglu,sevim}@gantep.edu.tr

Abstract. Primary objective of this study is to show how fuzzy optimization models can be solved directly by employing metaheuristics and ranking methods without requiring a transformation into a crisp model. In this study, a fuzzy multi-item Economic Order Quantity (EOQ) model with two constraints is solved directly (without any transformation process) by employing three different fuzzy ranking functions and the Particle Swarm Optimization (PSO) metaheuristic. The parameters of the problem are defined as symmetric triangular fuzzy numbers. Having fuzzy parameters, the objective function values of the generated solution vectors also will be fuzzy numbers. Therefore, in the selection of the best solution vector, ranking of fuzzy numbers is used. Similarly, the feasibility of the constraints for the generated solution vectors will be determined via ranking of two fuzzy numbers. By this approach other fuzzy optimization problems can be solved without any transformation process.

Keywords: EOQ; fuzzy ranking functions; particle swarm optimization.

1 Introduction

Most of the real life problems and models contain linguistic and/or imprecise variables and constraints. The mentioned impreciseness in a system does not exist because of randomness but rather because of fuzziness. The classical procedures are generally not suitable (or easy) to handle linguistic terms or impreciseness in a given mathematical program; therefore the decision maker is usually forced to state the problem in precise mathematical terms. Fuzzy set theory gives an opportunity to handle linguistic terms and vagueness in real life systems.

Fuzzy set theory gives the ability to quantitatively and qualitatively model problems which involve vagueness and impreciseness. Zimmermann [1] identifies that fuzzy set theory can be used as language to model problems which contain fuzzy phenomena or relationships, as a tool to analyze such models in order to gain better insight into the problem and as an algorithmic tool to make solution procedures more stable or faster [2].

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 32–42, 2008.

In the literature, there are various studies on solving fuzzy mathematical programming (FMP) models. In a FMP model, all or some of the parameters can be defined as fuzzy numbers. For FMP models with various fuzzy parameters, different optimization algorithms were proposed. But, most of the solution approaches are based on the fuzzy decision concept which was proposed by Zimmermann [3]. Other common approach is to use fuzzy ranking procedures as a part of the solution mechanism for solving FMPs. In the literature, there are various studies in which different fuzzy ranking procedures are used for the solution of fuzzy mathematical models. In all of these studies, FMP models were first transformed into a crisp equivalent then solved by a classical solution approach.

Inventory management is very important for many service and manufacturing industries. A proper control of inventory can significantly enhance a company's profitability. The purpose of the EOQ model is to find the optimal order quantity of inventory items at each time such that the combination of the order cost and the stock cost is minimal [4]. There are a variety of EOQ models available and all originate from the classical EOQ model [5]. In reality, it is very hard to define parameters of the EOQ model precisely. Moreover, it is very hard to estimate the probability distribution of these parameters due to a lack of historical data. Instead, these parameters are often estimated based on experience and subjective managerial judgment [4]. However, these non-stochastic and ill-formed inventory models can be realistically represented in the fuzzy environment [6]. In this study, a fuzzy multi-item EOQ model with two constraints (available warehouse space and number of orders placed during a time period) is handled. The parameters of the problem are defined as triangular fuzzy numbers. The fuzzy multi-item EOQ problem is solved directly by employing three different fuzzy ranking methods and the PSO metaheuristic algorithm. Ranking methods for fuzzy numbers are used to rank the objective function values and to determine the feasibility of the constraints. The aim of this study is to show that fuzzy models can be solved directly by using metaheuristics and ranking methods.

2 The Particle Swarm Optimization Algorithm

In order to solve the fuzzy multi-item EOQ problem directly by using the ranking methods, PSO is employed in this study. PSO is a simple algorithm that seems to be effective for optimizing a wide range of functions [7]. A PSO algorithm maintains a swarm of particles, where each particle represents a potential solution. A swarm is similar to a population, while a particle is similar to an individual. The particles are flown through a multidimensional search space, where the position of each particle is adjusted according to its own experience and that of its neighbors [8]. Two PSO algorithms have been developed which differ in the size of their neighborhoods; global best and local best PSO. For the global best PSO, the neighborhood for each particle is the entire swarm [8]. In local best PSO, particles have information only of their own and their nearest array neighbours' bests, rather than that of the entire group [7]. In this study,

global best PSO algorithm is used. Let $x_i(t)$denote the position of particle i in the search space at the time step t. The position of the particle is changed by adding a velocity, $v_i(t)$, to the current position,

$$x_i;(t+1) = x_i(t) + v_i(t+1) \tag{1}$$

It is the velocity that drives the optimization process, and reflects both the experiential knowledge of the particle and socially exchanged information from the particle's neighborhood [8]. The PSO concept consists of, at each time step, changing the velocity of each particle toward its particle best and global best. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward particle best and global best [7]. For global best PSO, the velocity of particle i is calculated as [8];

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2r_{2j}(t)[\widehat{y}_j(t) - x_{ij}(t)] \tag{2}$$

where $v_{ij}(t)$ is the velocity of particle i in dimension j=1,...,n_x at time step t, $x_{ij}(t)$is the position of particle i in dimension j at time step t, w is the inertia weight, c_1 and c_2 positive acceleration constants, $r_{1j}(t)$ and $r_{2j}(t)$ ˜U(0,1) are random values in the range [0,1], $y_{ij}(t)$ is the personal best position of particle i, $\widehat{y}_j(t)$is the global best position at time step t. Inertia weight w controls the impact of previous historical values of particle velocity on its current one. A larger inertia weight pressures toward global exploration while a smaller inertia weight pressures toward fine-tuning the current search area. The acceleration constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle towards *pbest* and *gbest* positions. Thus, adjustment of these constants changes the amount of tension in the system [9]. The global best PSO algorithm can be summarized as follows [8].

Create and initialize an n_x-dimensional swarm, S;
repeat
  for each particle $i = 1,\ldots,S.ns$ **do**
    // set the personal best position
    if $f(S.xi) < f(S.yi)$ **then**
    $S.y_i = S.x_i$;
    end
    // set the global best position
    if $f(S.y_i) < f(S.\widehat{y})$**then**
    $S.\widehat{y} = S.y_i$;
    end
  end
  for each particle $i = 1,\ldots,S.ns$ **do**
  update the velocity using eq. 2;
  update the position using eq. 1;
  end
until stopping condition is true.

In order to solve the fuzzy multi-item EOQ problem directly, the above global best PSO algorithm is used. In the solution of the problem, the parameters of the algorithm are taken as follows after several trial runs; inertia weight $w = 0.4$, individual and sociality weights $c_1 = c_2 = 1.4962$, and the number of iterations is 1000.

3 Fuzzy Multi-item EOQ Model

EOQ models are used for determining the quantity of item(s) to purchase from suppliers or to process through a production facility [5]. Inventory management is used to decide when and how much to replenish the companies' inventory under a minimum of total cost. An EOQ model can be defined as only the minimization of the cost function or minimization of cost function under limitations like that budget, warehouse space, number of orders, etc. In this study, a multi-item EOQ model with two constraints (available warehouse space and number of orders placed during a time period) is handled. The problem is to decide the order levels Q_i, $i = 1, 2, \ldots, m$ which minimize the average total cost. In this study, the fuzzy multi-item EOQ problem is solved by employing three different fuzzy ranking methods and the PSO algorithm. The studied fuzzy multi-item EOQ model is defined as follows;

$$\begin{aligned} \min \quad & C(Q) = \sum_{1}^{m} \left(\frac{\widetilde{c}_{1i} Q_i}{2} + \frac{\widetilde{c}_{2i} D_i}{Q_i} \right) \\ \textit{subject to} \quad & \sum_{1}^{m} \widetilde{a}_i Q_i \leq \widetilde{W}; \qquad \sum_{1}^{m} \frac{\widetilde{M}_i}{Q_i} \leq \widetilde{n}; \quad Q \geq 0 \end{aligned} \tag{3}$$

where, $Q_i$is the economic order quantity for i^{th} item, $\widetilde{c}_{1i}$ is the holding cost per unit quantity per unit time for i^{th} item, $\widetilde{c}_{2i}$ is the set up cost per period for i^{th} item, $D_i$is the demand per unit time for i^{th} item, $\widetilde{a}_i$ is the space required by each unit of product i (in sq.m), $\widetilde{M}_i$ is the total demand of product i during some given time interval, $\widetilde{W}$ is the maximum available warehouse space (in sq.m.), $\widetilde{n}$ is the maximum number of orders placed during the given time period and m is the number of items. The parameters of the problem are defined as triangular fuzzy numbers. As an example application, it is accepted that there are two items in the EOQ problem. The input data of the example problem is given in Table 1 and Table 2. The input data is similar to the data that was used by Mondal and Maiti's [10] multi-item EOQ problem, except and which are not considered as fuzzy numbers in their work. In the study of Mondal and Maiti [10], the objective function, the cost coefficients and the right hand values of the constraints were defined as fuzzy numbers. Moreover, they did not employ triangular fuzzy numbers and set aspiration value for the objective function. Therefore the present model is different than Mondal and Maiti's model.

Table 1. Input data relevant to items

Item	$\widetilde{c}_{1i}$	$\widetilde{c}_{2i}$	D_i	$\widetilde{a}_i$	$\widetilde{M}_i$
1	(200;250;300)	$(90.10^3; 10^5; 110.10^3)$	200	(0.8;1;1.1)	(7500;8000;8500)
2	(150;200;250)	$(225.10^3; 245.10^3; 265.10^3$	800	(0.8;1;1.1)	(3500;4000;4500)

Table 2. Input data relevant to production environment

$\widetilde{W} = (\underline{W}; W; \overline{W})$	$\widetilde{n} = (\underline{n}; n; \overline{n})$
(1450; 1500; 1550)	(18; 20; 22)

The fuzzy multi-item EOQ problem of the present study can be stated as follows;

$$\begin{aligned} \min \quad & C(Q) = (200;250;300)\frac{Q_1}{2} + (90.10^3; 10^5; 110.10^3)\frac{200}{Q_1} + \\ & (150;200;250)\frac{Q_2}{2} + (225.10^3; 245.10^3; 265.10^3)\frac{800}{Q_2} \\ \textit{subject to} \quad & (0.8;1;1.1)Q_1 + (0.8;1;1.1)Q_2 \leq (1450;1500;1550) \\ & (7500;8000;8500)\frac{1}{Q_1} + (3500;4000;4500)\frac{1}{Q_2} \leq (18;20;22) \\ & Q_1, Q_2 \geq 0 \end{aligned} \tag{4}$$

In the present study, the fuzzy multi-item EOQ problem is solved directly by using three different ranking methods and the PSO algorithm. Ranking methods for fuzzy numbers are used to rank the objective function values and to determine the feasibility of the constraints. As the cost coefficients of the objective functions are fuzzy numbers, the objective function values of the generated solution vectors will be fuzzy numbers. Therefore, in the selection of the best solution vector, ranking of fuzzy numbers is used. Similarly, the feasibility of the constraints for the generated solution vectors will be determined via ranking of two fuzzy numbers (i.e. comparing right and left hand side fuzzy numbers for the constraint functions). In the following section, both the solution of the problem with transformation process and the solution of the problem with the proposed direct approach are shown.

3.1 Solution of Fuzzy Multi-item EOQ Problem Via the Signed Distance Method

Yao and Wu [11] have used signed distance to define ranking of fuzzy numbers. The signed distance used for fuzzy numbers has some similar properties to the properties induced by the signed distance in real numbers. Let F be the family

of the fuzzy numbers on R. The sign distance is defined as $d^*(a,0) = a$ on R. Then for $a, b \in R$, $d^*(a,b) = a - b$. For , $\widetilde{D}, \widetilde{E} \in F$, with α-cut $(0 \leq \alpha \leq 1)$, there is a closed interval $D(\alpha) = [D_L(\alpha), D_R(\alpha)]$. Then, the signed distance of $\widetilde{D}, \widetilde{E}$, is defined as [11],

$$d(\widetilde{D}, \widetilde{E}) = \frac{1}{2}\int_0^1 [D_L(\alpha) + D_R(\alpha) - E_L(\alpha) - E_R(\alpha)]d\alpha \tag{5}$$

It can be proved that d is an extension of d^*. According to these definitions, the signed distance of a triangular fuzzy number $\widetilde{A} = (\underline{a}; a; \overline{a})$is defined as,

$$d(\widetilde{A}, 0) = \frac{1}{2}\int_0^1 [\underline{a} + (a - \underline{a})\alpha + \overline{a} - (\overline{a} - a)\alpha]d\alpha = \frac{1}{4}(2a + \underline{a} + \overline{a}) \tag{6}$$

Let and are two triangular fuzzy numbers, their ranking relation is defined as $\widetilde{A} \leq \widetilde{B} \iff d(\widetilde{A}, 0) \leq d(\widetilde{B}, 0)$ [11].

In the study of Baykasoğlu and Göçken [12], the fuzzy EOQ problem (eq. 4) is solved after transformed into crisp equivalent. The resultant crisp nonlinear problem is solved by using LINGO solver. The solution obtained from LINGO is $Q_1 = 494.8279, Q_2 = 1043.634$, and the corresponding objective function value of the crisp model is 394440.3. Triangular possibility distribution of the objective function is determined after finding the optimal solution vector. The cost coefficients of the problem are triangular fuzzy numbers, so by calculating the objective function value by using minimum, middle and maximum points of the cost coefficients separately, the possibility distribution of the objective function can be obtained. The triangular possibility distribution of the objective function for the obtained solution using signed distance method is (336605.88; 394440.28; 452274.69). In this study, the fuzzy multi-item EOQ problem is solved directly by using the signed distance method and the PSO algorithm. The signed distance method is used to rank the objective function values and to determine the feasibility of the constraints. The obtained solution is , $Q_1 = 494.8279, Q_2 =$ 1043.634, z = (336605.92; 394440.32; 452274.72). The solution obtained after transformation process and solution obtained from direct solution approach are same. However the need for transformation is avoided in the present approach.

3.2 Solution of Fuzzy Multi-item EOQ Problem Via Ranking with Integral Value

Liou and Wang [13] proposed the method of ranking fuzzy numbers with integral value. Ranking fuzzy numbers with integral value is relatively simple in computation, especially in ranking of triangular and trapezoidal fuzzy numbers, and can be used to rank more than two fuzzy numbers simultaneously [14].

Table 3. α-acceptable optimal solutions for the integral value method

Feasibility degree, α	Decision vector, $Q^0(\alpha)$	*Possibility distribution of the objective value*, $\widetilde{z}^0(\alpha)$
0.4	$Q_1 = 494.2751$ $Q_2 = 1063.017$	(334900.12; 392930.26; 450960.39)
0.5	$Q_1 = 494.8279$ $Q_2 = 1043.634$	(336605.88; 394440.28; 452274.69)
0.6	$Q_1 = 495.4511$ $Q_2 = 1024.751$	(338384.39; 396039.72; 453695.04)
0.7	$Q_1 = 496.1461$ $Q_2 = 1006.341$	(340235.63; 397728.06; 455220.49)
0.8	$Q_1 = 496.9146$ $Q_2 = 988.3795$	(342159.73; 399505.04; 456850.34)
0.9	$Q_1 = 497.7585$ $Q_2 = 970.8405$	(344157.36; 401370.91; 458584.46)
1.0	$Q_1 = 498.6800$ $Q_2 = 953.7010$	(346229.27; 403326.12; 460422.99)

The definition of integral values for the triangular fuzzy number is defined as follows [14].

$$I(\widetilde{A}) = \frac{1-\alpha}{2}\underline{a} + \frac{1}{2}a + \frac{\alpha}{2}\overline{a} \tag{7}$$

where $0 \leq \alpha \leq 1$. The index of optimism α is representing the degree of optimism for a person. A larger α indicates a higher degree of optimism [14]. The fuzzy numbers are ranked according to their integral values; the fuzzy number with the larger integral value is the bigger fuzzy number.

When the Fuzzy EOQ problem is transformed into crisp equivalent using the integral value method, an α parametric nonlinear crisp problem is obtained and it has solved by using LINGO solver in the study of Baykasoğlu and Göçken [12]. The resultant crisp problem **??** is solved for different α values (α values are determined by the decision maker). The multi-item EOQ problem is solved for $\alpha = \{0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1.0\}$. The obtained solutions and the possibility distributions of the objective for each α value are given in Table 3 [12]. In the present study, the fuzzy multi-item EOQ problem is solved directly by using ranking fuzzy numbers with integral value and the PSO algorithm. The integral value ranking method is used to rank the objective function values and to determine the feasibility of the constraints. The obtained solutions for each α value are given in Table 4. As it can be seen from Table 3 and Table 4 the obtained solutions from transformation approach and direct solution approach are exactly the same.

3.3 Solution of Fuzzy Multi-item EOQ Problem Via Ranking of Fuzzy Numbers through the Comparison of Their Expected Intervals

Jimenez [15] has proposed a ranking method of fuzzy numbers based on the comparison of their expected intervals. If a fuzzy number is triangular, its expected interval will be [15]:

$$EI(\widetilde{A}) = [E_1^{\widetilde{A}}, E_2^{\widetilde{A}}] = [\frac{1}{2}(\underline{a} + a), \frac{1}{2}(a + \overline{a})] \tag{8}$$

Table 4. α-acceptable optimal solutions for the integral value method with PSO algorithm

Feasibility degree, α	Decision vector, $Q^0(\alpha)$	*Possibility distribution of the objective value*, $\widetilde{z}^0(\alpha)$
0.4	$Q_1 = 494.2751 Q_2 = 1063.017$	(334900.12; 392930.26; 450960.39)
0.5	$Q_1 = 494.8279 \quad Q_2 = 1043.634$	(336605.92; 394440.32; 452274.72)
0.6	$Q_1 = 495.4511 \quad Q_2 = 1024.751$	(338384.40; 396039.73; 453695.05)
0.7	$Q_1 = 496.1461 \quad Q_2 = 1006.341$	(340235.60; 397728.03; 455220.47)
0.8	$Q_1 = 496.9146 \quad Q_2 = 988.3795$	(342159.74; 399505.04; 456850.34)
0.9	$Q_1 = 497.7585 \quad Q_2 = 970.8405$	(344157.36; 401370.92; 458584.47)
1.0	$Q_1 = 498.6800 \quad Q_2 = 953.7010$	(346229.27; 403326.13; 460422.99)

According to the ranking method of Jimenez, for any pair of fuzzy numbers and, the degree in which is bigger than is defined as [15,16];

$$\mu_M(\widetilde{A},\widetilde{B}) = \left\{ \begin{array}{c} 0, \quad if \quad E_2^{\widetilde{A}} - E_1^{\widetilde{B}} < 0 \\ \frac{E_2^{\widetilde{A}} - E_1^{\widetilde{B}}}{E_2^{\widetilde{A}} - E_1^{\widetilde{B}} - (E_1^{\widetilde{A}} - E_2^{\widetilde{B}})}, \quad if \quad 0 \in [E_1^{\widetilde{A}} - E_2^{\widetilde{B}}, E_2^{\widetilde{A}} - E_1^{\widetilde{B}}] \\ 1, \quad if \quad E_1^{\widetilde{A}} - E_2^{\widetilde{B}} > 0 \end{array} \right\} \tag{9}$$

Where, $\mu_M(\widetilde{A},\widetilde{B})$ is the degree of preference of $\widetilde{A}$ over $\widetilde{B}$. When $\mu_M(\widetilde{A},\widetilde{B}) = 0.5$ it will be said that and $\widetilde{A}$ and $\widetilde{B}$ are equal. The expected value of a fuzzy number is the half point of its expected interval:

$$EV(\widetilde{A}) = \frac{E_1^{\widetilde{A}} + E_2^{\widetilde{A}}}{2} \tag{10}$$

Jimenez et al. [16] have used the expected values of fuzzy numbers and the ranking of fuzzy numbers using expected intervals for solving FMP problems in which all parameters are defined as fuzzy numbers. Based on Jimenez's approach, a FMP problem is transformed into an equivalent α-parametric crisp problem. In the study of Baykasoğlu and Göçken [12], the equivalent α-parametric crisp nonlinear problem of the fuzzy EOQ problem is obtained and solved by using LINGO solver. The crisp model is solved for $\alpha = \{0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1.0\}$. The obtained solutions and the possibility distributions of the objective for each α value are given in Table 5.

In the present study, the fuzzy multi-item EOQ problem is solved directly via ranking of fuzzy numbers through the comparison of their expected intervals and the PSO algorithm. The ranking method is used to rank the objective function values and to determine the feasibility of the constraints. The obtained solutions for each α value are given in Table 6.

As it can be seen from Tables 5 and 6 the results are different, except α=0.5. This is mainly due to the fact that; in Jimenez's transformation approach, the best objective value is the objective value which is a better choice at least in degree 0.5 as opposed to the others. But, in deciding the feasibility of the constraints,

Table 5. α-acceptable optimal solutions for the expected intervals method

Feasibility degree, α	Decision vector, $Q^0(\alpha)$	*Possibility distribution of objective value*, $\widetilde{z}^0(\alpha)$
0.4	$Q_1 = 479.8072 \quad Q_2 = 1087.901$	(332544.6; 390612.9; 448681.13)
0.5	$Q_1 = 494.8279 \quad Q_2 = 1043.634$	(336605.92; 394440.32; 452274.72)
0.6	$Q_1 = 511.2642 \quad Q_2 = 998.8368$	(341455.65; 399138.68; 456821.71)
0.7	$Q_1 = 529.4697 \quad Q_2 = 953.1174$	(347281.03; 404910.09; 462539.15)
0.8	$Q_1 = 549.9703 \quad Q_2 = 905.912$	(354364.28; 412059.66; 469755.04)
0.9	$Q_1 = 573.6106 \quad Q_2 = 856.341$	(363163.45; 421083.07; 479002.69)
1.0	$Q_1 = 601.9098 \quad Q_2 = 802.852$	(374510.42; 432881.18; 491251.93)

Table 6. α-acceptable optimal solutions for the expected intervals method with PSO algorithm

Feasibility degree, α	Decision vector, $Q^0(\alpha)$	*Possibility distribution of objective value*, $\widetilde{z}^0(\alpha)$
0.4	$Q_1 = 488.6419 \quad Q_2 = 1010.0089$	(339667.89; 397068.59; 454469.28)
0.5	$Q_1 = 494.8279 \quad Q_2 = 1043.634$	(336605.92; 394440.32; 452274.72)
0.6	$Q_1 = 514.7356 \quad Q_2 = 994.3181$	(342045.41; 399748.68; 457451.94)
0.7	$Q_1 = 584.1816 \quad Q_2 = 872.1778$	(361023.77; 419201.23; 477378.69)
0.8	$Q_1 = 644.8545 \quad Q_2 = 724.4589$	(395194.43; 454614.19; 514033.95)
0.9	$Q_1 = 644.8545 \quad Q_2 = 724.4589$	(395194.43; 454614.19; 514033.95)
1.0	$Q_1 = 644.8545 \quad Q_2 = 724.4589$	(395194.43; 454614.19; 514033.95)

different ordering degrees are used after transforming into crisp equivalent. In our approach, the selection of the best objective value and the feasibility check of constraints are carried out at the same ordering degrees. The best objective function value is decided according to the ordering degrees between 0.5 and 1.0 as in deciding the feasibility of the constraints. When the ordering degree which is used in the selection of the best objective function value increases, the obtained objective function value is expected to increase as well. Therefore, objective function values generated from the direct solution can be bigger than the objective function values generated from the transformation process for different ordering degrees.

4 Conclusion

The aim of this study is to present that fuzzy optimization models can be solved directly by employing metaheuristics and ranking methods without requiring a transformation into a crisp model. For this purpose a fuzzy multi-item EOQ model with two constraints is handled. The parameters of the problem are defined as triangular fuzzy numbers. The fuzzy multi-item EOQ problem is solved directly by employing three different fuzzy ranking methods and the PSO algorithm. Both the solution of the problem with transformation process and the solution of the problem with the proposed direct solution are presented and

compared. It is seen that same results can be obtained from solution with transformation process and direct solution approach. According to this, it has been observed that FMP problems can be solved effectively by using ranking methods of fuzzy numbers without any necessity of transformation into crisp equivalent. Essentially, it can be very hard to transform many problems into crisp equivalent and sometimes the obtained crisp equivalent can be highly nonlinear. When the obtained crisp equivalent is nonlinear, a meta-heuristics algorithm should be used again for the solution. Therefore, transformation might not be always advantageous; in fact it can be unnecessary.

Acknowledgments

The first author is grateful to Turkish Academy of Sciences (TÜBA) for supporting his scientific studies. The second author is grateful to TUBITAK for supporting her PhD work.

References

1. Zimmermann, H.-J.: Using Fuzzy Sets in Operational Research. European Journal of Operational Research 13, 201–216 (1983)
2. Guiffrida, A.L., Nagi, R.: Fuzzy Set Theory Applications in Production Management Research: A Literature Survey. Journal of Intelligent Manufacturing 9, 39–56 (1998)
3. Zimmermann, H.J.: Description and Optimization of Fuzzy Systems. International Journal of General Systems 2, 209–215 (1976)
4. Wang, X., Tang, W., Zhao, R.: Fuzzy Economic Order Quantity Inventory Models without Backordering. Tsinghua Science and Technology 12, 91–96 (2007)
5. Stockton, D.J., Quinn, L.: Identifying Economic Order Quantities Using Genetic Algorithms. International Journal of Operations & Production Management 13, 92–103 (1993)
6. Roy, T.K., Maiti, M.: Multi-Objective Inventory Models of Deteriorating Items with Some Constraints in a Fuzzy Environment. Computers and Operations Research 25, 1085–1095 (1998)
7. Eberhart, R., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: The Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
8. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley & Sons Ltd., England (2005)
9. Dong, Y., Tang, J., Xu, B., Wang, D.: An Application of Swarm Optimization to Nonlinear Programming. Computers and Mathematics with Applications 49, 1655–1668 (2005)
10. Mondal, S., Maiti, M.: Multi-Item Fuzzy EOQ Models Using Genetic Algorithm. Computers and Industrial Engineering 44, 105–117 (2002)
11. Yao, J.-S., Wu, K.: Ranking Fuzzy Numbers based on Decomposition Principle and Signed Distance. Fuzzy Sets and Systems 116, 275–288 (2000)
12. Baykasoğlu, A., Göçken, T.: Solution of a Fully Fuzzy Multi-Item Economic Order Quantity Problem by Using Fuzzy Ranking Functions. Engineering Optimization 39, 919–939 (2007)

13. Liou, T.-S., Wang, M.J.: Ranking Fuzzy Numbers with Integral Value. Fuzzy Sets and Systems 50, 247–255 (1992)
14. Liou, T.-S., Chen, C.-W.: Fuzzy Decision Analysis for Alternative Selection Using a Fuzzy Annual Worth Criterion. The Engineering Economist 51, 19–34 (2006)
15. Jimenez, M.: Ranking Fuzzy Numbers through the Comparison of Its Expected Intervals. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 4, 379–388 (1996)
16. Jimenez, M., Arenas, M., Bilbao, A., Rodriguez, M.V.: Linear Programming with Fuzzy Parameters: An Interactive Method Resolution. Eur. J. of Oper. Research 177, 1599–1609 (2007)

Linear Reformulations of Integer Quadratic Programs

Alain Billionnet[1], Sourour Elloumi[2], and Amélie Lambert[2]

[1] CEDRIC-ENSIIE, 18 allée Jean Rostand, F-91025 Evry cedex, France
alain.billionnet@ensiie.fr
[2] CEDRIC-CNAM, 292 rue Saint-Martin, F-75141 Paris cedex 03, France
sourour.elloumi@cnam.fr, amelie.lambert@cnam.fr

Abstract. Let (QP) be an integer quadratic program that consists in minimizing a quadratic function subject to linear constraints. In this paper, we present several linearizations of (QP). Many linearization methods for the quadratic 0-1 programs are known. A natural approach when considering (QP) is to reformulate it into a quadratic 0-1 program. However, this method, that we denote `BBL` (Binary Binary Linearization), leads to a quadratic program with a large number of variables and constraints.

Our new approach, `BIL` (Binary Integer Linearization), consists in reformulating (QP) into a particular quadratic integer program where each quadratic term is the product of an integer variable by a 0-1 variable. The obtained integer linear program is significantly smaller than in the `BBL` approach.

Each reformulation leads to an integer linear program that we improve by adding valid inequalities. Finally, we get 4 different programs that we compare from the computational point of view.

Keywords: Integer programming, quadratic programming, linear reformulations.

1 Introduction

Consider the following linearly-constrained integer quadratic program:

$$(QP)\begin{cases} Min\ f(x) = x^T Q x + c^T x \\ s.t \quad x \in X \subset \mathbb{N}^n \end{cases}$$

with $Q \in \mathbf{S}_n$ (space of symmetric matrices of order n), $c \in \mathbb{R}^n$ and X is defined as the set of integer solutions of a system of linear equalities and inequalities:

$$X = \left\{ x : \begin{array}{lll} Ax = b & & (1) \\ Dx \leq e & & (2) \\ x_i \leq u_i & i \in I & (3) \\ x_i \geq 0 & i \in I & (4) \\ x_i \in \mathbb{N} & i \in I & (5) \end{array} \right.$$

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 43–51, 2008.

where $A \in \mathbf{M}_{m,n}$ (set of $m * n$ integer matrices), $b \in \mathbb{N}^m$, $D \in \mathbf{M}_{p,n}$, $e \in \mathbb{N}^p$, $u \in \mathbb{N}^n$, $I = \{i : i = 1, \ldots, n\}$. Without loss of generality, we shall suppose X non empty.

We denote $R = \{r : r = 1, \ldots, m\}$, $S = \{s : s = 1, \ldots, p\}$, $E = \{(i,k) : i = 1, \ldots, n,\ k = 0, \ldots \lfloor log(u_i) \rfloor\}$ and $N = |E| = \sum_{i=1}^{n} (\lfloor log(u_i) \rfloor + 1)$.

A lot of applications in operations research and industrial engineering involve discrete variables in their formulation. Some of these applications can be formulated as (QP). For instance, such a formulation is used in (1) for the chaotic mapping of complete multipartite graphs.

In the state-of-the-art, a majority of resolution methods of quadratic discrete problems are designed only for quadratic 0-1 programs. This is why a natural way to solve (QP) consists in replacing each integer variable by its binary decomposition. The number of additional variables is hence equal to N. Thereafter each integer product becomes an expression of binary products, that we standardly linearize. The idea of the standard 0-1 linearization (2) consists in adding a set of new variables and a family of inequalities that we substitute to the binary quadratic terms. The main drawback of this approach, that we call `BBL` (Binary Binary Linearization) is that the size of the obtained linear problem is $O(N^2)$. Possible improvements of the standard 0-1 linearization were introduced by Sherali and Adams (3) and consist in adding a family of valid inequalities. These improvements can be easily applied to the `BBL` approach, giving a reinforced linearization method that we call `BBLr`.

Our new approach, that we call `BIL` (Binary Integer Linearization), consists also in replacing each integer variable by its binary decomposition. Then, in each product of two different integer variables we replace only one of them by its binary decomposition. Thus, each integer product becomes an expression of products of a binary variable by an integer one. Finally, we linearize these new products by the standard binary-integer linearization (4). The `BIL` approach hence leads to an integer linear program of size $O(nN)$ that is significantly smaller than the program of size $O(N^2)$ provided by the `BBL` method. Moreover, we improve this reformulation in term of integrality gap, by adding new valid inequalities. We denote by `BILr` the reinforced version of the `BIL` method.

Finally, we get 4 linear reformulations that we compare from the computational point of view. Our experimentations are carried out on the Integer Quadratic Knapsack Problem (IQKP).

The paper is organized as follows. In Section 2, we present the `BBL` approach and its reinforcement `BBLr`. In Section 3, we describe the `BIL` approach and its reinforcement `BILr`. Finally, in Section 4, we present our computational study of these different methods. Section 5 is a conclusion.

2 The BBL Approach

Let $x_i = \sum_{k=0}^{\lfloor log(u_i) \rfloor} 2^k t_{ik}$ be the unique binary decomposition of x_i. We replace the x_i variables by the set of t_{ik} binary variables. Then each product $x_i x_j$ leads to an expression of products $t_{ik} t_{jl}$, that we linearize by adding new binary variables y_{ikjl}. We obtain the following program:

$$
(LP_{\texttt{BBL}}) \begin{cases}
Min\ f_{\texttt{BBL}}(x,y) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ q_{ij} \neq 0}}^{n} q_{ij} \sum_{k=0}^{\lfloor log(u_i) \rfloor} \sum_{l=0}^{\lfloor log(u_j) \rfloor} 2^{k+l} y_{ikjl} + \sum_{i=1}^{n} c_i x_i & \\
s.t. \quad (1)(2)(3) & \\
\qquad x_i = \sum_{k=0}^{\lfloor log(u_i) \rfloor} 2^k t_{ik} \quad i \in I & (6) \\
\qquad y_{ikjl} \leq t_{ik} \qquad (i,k),(j,l) \in E,\ q_{ij} < 0 & (7) \\
\qquad y_{ikjl} \leq t_{jl} \qquad (i,k),(j,l) \in E,\ q_{ij} < 0 & (8) \\
\qquad y_{ikjl} \geq t_{ik} + t_{jl} - 1 \quad (i,k),(j,l) \in E,\ q_{ij} > 0 & (9) \\
\qquad y_{ikjl} \geq 0 \qquad (i,k),(j,l) \in E,\ q_{ij} > 0 & (10) \\
\qquad y_{ikjl} = y_{jlik} \qquad (i,k),(j,l) \in E,\ i < j\ q_{ij} \neq 0 & (11) \\
\qquad y_{ikik} = t_{ik} \qquad (i,k) \in E,\ q_{ii} \neq 0 & (12) \\
\qquad y_{ikil} = y_{ilik} \qquad (i,k),(i,l) \in E,\ k < l,\ q_{ii} \neq 0 & (13) \\
\qquad t_{ik} \in \{0,1\} \qquad (i,k) \in E & (14)
\end{cases}
$$

Observe that for any optimal solution of $(LP_{\texttt{BBL}})$, as variables y_{ikjl} are present only in the objective function and in Constraints (7)-(13), the following properties are satisfied:

- If $q_{ij} < 0$ then $y_{ikjl} = \min(t_{ik}, t_{jl})$
- If $q_{ij} > 0$ then $y_{ikjl} = \max(0, t_{ik} + t_{jl} - 1)$

ensuring y_{ikjl} to be equal to the product $t_{ik} t_{jl}$ if Constraints (14) are satisfied. Constraints (11) and (13) follow from the equality $t_{ik} t_{jl} = t_{jl} t_{ik}$. Constraints (12) follow from the property that if $t_{ik} \in \{0,1\}$ then $t_{ik}^2 = t_{ik}$.

The size of $(LP_{\texttt{BBL}})$ is $O(N^2)$. As the y_{ikjl} variables and related constraints are not defined when $q_{ij} = 0$, the actual size depends on the density of matrix Q. In our computational results of Section 4, matrix Q is fully dense.

Improving the BBL approach

Here we improve the BBL approach by adding valid inequalities in $(LP_{\texttt{BBL}})$ following the same ideas as in (3). We generate valid inequalities by multiplying the initial constraints (1) and (2) by the binary variables, then we linearize the obtained quadratic constraints. We obtain the following reinforced program:

$$(LP_{\texttt{BBLr}})\begin{cases} Min\ f_{\texttt{BBLr}}(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij} \sum_{k=0}^{\lfloor log(u_i)\rfloor}\sum_{l=0}^{\lfloor log(u_j)\rfloor} 2^{k+l} y_{ikjl} + \sum_{i=1}^{n} c_i x_i \\ s.t.\quad (1)(2)(3)(6)(14) \\ \quad y_{ikjl} \le t_{ik} & (i,k),(j,l) \in E & (7') \\ \quad y_{ikjl} \le t_{jl} & (i,k),(j,l) \in E & (8') \\ \quad y_{ikjl} \ge t_{ik} + t_{jl} - 1 & (i,k),(j,l) \in E & (9') \\ \quad y_{ikjl} \ge 0 & (i,k),(j,l) \in E & (10') \\ \quad y_{ikjl} = y_{jlik} & (i,k),(j,l) \in E,\ i<j & (11') \\ \quad y_{ikik} = t_{ik} & (i,k) \in E & (12') \\ \quad y_{ikil} = y_{ilik} & (i,k),(i,l) \in E,\ k<l & (13') \\ \quad \sum_{i=1}^{n}\sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k a_{ri} y_{ikjl} = b_r t_{jl} & (j,l) \in E,\ r \in R & (15) \\ \quad \sum_{i=1}^{n}\sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k d_{si} y_{ikjl} \le e_s t_{jl} & (j,l) \in E,\ s \in S & (16) \\ \quad \sum_{i=1}^{n} d_{si} x_i - \sum_{i=1}^{n}\sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k d_{si} y_{ikjl} \le e_s(1-t_{jl}) & (j,l) \in E,\ s \in S & (17) \end{cases}$$

We multiply the equality Constraints (1) by variable t_{jl} to get Constraints (15). Similarly, we multiply the inequality Constraints (2) by t_{jl} (resp. $(1-t_{jl})$) to get Constraints (16) (resp. (17)). Doing this introduces variables y_{ikjl} in the new constraints (15)-(17). Hence we need to define Constraints (7')-(13') independently from the sign of q_{ij}. Moreover, variables y_{ikjl} become needed even when $q_{ij} = 0$. The size of $(LP_{\texttt{BBLr}})$ does no longer depend on the density of matrix Q.

3 The BIL Approach

Here again we use the unique binary decomposition $x_i = \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k t_{ik}$. We linearize the square terms x_i^2 by use of variables y_{ikil} that represent the product $t_{ik}t_{il}$ as in the BBL approach. However, for quadratic terms $x_i x_j$ with $i \neq j$, we use the equality $x_i x_j = \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k t_{ik} x_j$, that we linearize by introducing new variables z_{ijk} to replace each quadratic term $t_{ik}x_j$. Then we add a set of inequalities that ensure z_{ijk} to be equal to $t_{ik}x_j$. We obtain the following program:

$$(LP_{\texttt{BIL}})\begin{cases} Min\ f_{\texttt{BIL}}(x,y,z) \\ s.t\quad (1)(2)(3)(6)(14) \\ \quad z_{ijk} \le u_j t_{ik} & (i,k) \in E,\ j \in I,\ q_{ij} < 0,\ i \neq j & (18) \\ \quad z_{ijk} \le x_j & (i,k) \in E,\ j \in I,\ q_{ij} < 0,\ i \neq j & (19) \\ \quad z_{ijk} \ge x_j - u_j(1-t_{ik}) & (i,k) \in E,\ j \in I,\ q_{ij} > 0,\ i \neq j & (20) \\ \quad z_{ijk} \ge 0 & (i,k) \in E,\ j \in I,\ q_{ij} > 0,\ i \neq j & (21) \\ \quad y_{ikik} = t_{ik} & (i,k) \in E,\ q_{ii} \neq 0 & (22) \\ \quad y_{ikil} = y_{ilik} & (i,k),(i,l) \in E,\ k<l,\ q_{ii} \neq 0 & (23) \\ \quad y_{ikil} \le t_{ik} & (i,k),(i,l) \in E,\ q_{ii} < 0 & (24) \\ \quad y_{ikil} \le t_{il} & (i,k),(i,l) \in E,\ q_{ii} < 0 & (25) \\ \quad y_{ikil} \ge t_{ik} + t_{il} - 1 & (i,k),(i,l) \in E,\ q_{ii} > 0 & (26) \\ \quad y_{ikil} \ge 0 & (i,k),(i,l) \in E,\ q_{ii} > 0 & (27) \end{cases}$$

with

$$f_{\texttt{BIL}}(x,y,z) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ q_{ij}\neq 0 \\ i\neq j}}^{n} q_{ij} \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{ijk} + \sum_{i=1}^{n} c_i x_i + \sum_{\substack{i=1 \\ q_{ii}\neq 0}}^{n} q_{ii} \sum_{k=0}^{\lfloor log(u_i)\rfloor} \sum_{l=0}^{\lfloor log(u_i)\rfloor} 2^{k+l} y_{ikil}$$

In any optimal solution of program $(LP_{\texttt{BIL}})$ we have:

- If $q_{ij} < 0$ then $z_{ijk} = \min(u_j t_{ik}, x_j)$
- If $q_{ij} > 0$ then $z_{ijk} = \max(0, u_j t_{ik} + x_j - u_j)$

it follows that, if $t_{ik} = 0$ then $z_{ijk} = 0$ and if $t_{ik} = 1$ then $z_{ijk} = x_j$. This proves that in any optimal integer solution, $z_{ijk} = t_{ik}x_j$. For the same reason as for program $(LP_{\texttt{BBL}})$ we also have $y_{ikil} = t_{ik}t_{il}$. Hence program $(LP_{\texttt{BIL}})$ is a mixed integer linear program that is equivalent to (QP).

The BIL approach produces program $(LP_{\texttt{BIL}})$ with $O(nN)$ variables and constraints. Here again, it is not necessary to define z_{ijk} when $q_{ij} = 0$. The actual size depends on the density of matrix Q.

Improving the BIL approach

We mainly add Constraints (28)-(35) and variables z_{iik} that represent $t_{ik}x_i$. We also need to transform Constraints (18)-(27) into Constraints (18')-(27'). All this give the following integer linear program $(LP_{\texttt{BILr}})$.

$$(LP_{\texttt{BILr}}) \begin{cases} Min\ f_{\texttt{BILr}}(x,z) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i\neq j}}^{n} q_{ij} \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{ijk} + \sum_{i=1}^{n} q_{ii} \sum_{k=0}^{\lfloor log(u_i)\rfloor} \sum_{l=0}^{\lfloor log(u_i)\rfloor} 2^{k+l} y_{ikil} + \sum_{i=1}^{n} c_i x_i & \\ s.t \quad (1)(2)(3)(6)(14) & \\ z_{ijk} \le u_j t_{ik} & (i,k)\in E,\ j\in I \quad (18') \\ z_{ijk} \le x_j & (i,k)\in E,\ j\in I \quad (19') \\ z_{ijk} \ge x_j - u_j(1-t_{ik}) & (i,k)\in E,\ j\in I \quad (20') \\ z_{ijk} \ge 0 & (i,k)\in E,\ j\in I \quad (21') \\ y_{ikik} = t_{ik} & (i,k)\in E \quad (22') \\ y_{ikil} = y_{ilik} & (i,k),(i,l)\in E,\ k<l \quad (23') \\ y_{ikil} \le t_{ik} & (i,k),(i,l)\in E \quad (24') \\ y_{ikil} \le t_{il} & (i,k),(i,l)\in E \quad (25') \\ y_{ikil} \ge t_{ik} + t_{il} - 1 & (i,k),(i,l)\in E \quad (26') \\ y_{ikil} \ge 0 & (i,k),(i,l)\in E \quad (27') \\ \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{ijk} = \sum_{l=0}^{\lfloor log(u_j)\rfloor} 2^l z_{jil} & i,j\in I \quad (28) \\ \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{ijk} \ge x_i u_j + x_j u_i - u_i u_j & (i,k)\in E,\ j\in I \quad (29) \\ z_{iik} = \sum_{l=0}^{\lfloor log(u_i)\rfloor} 2^l y_{ikil} & (i,k)\in E \quad (30) \\ \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{iik} \ge x_i & i\in I \quad (31) \\ \sum_{i=1}^{n} a_{ri} z_{jil} = b_r t_{jl} & (j,l)\in E,\ r\in R \quad (32) \\ \sum_{i=1}^{n} d_{si} z_{jil} \le e_s t_{jl} & (j,l)\in E,\ s\in S \quad (33) \\ \sum_{i=1}^{n} (d_{si}x_i - d_{si}z_{jil}) \le e_s(1-t_{jl}) & (j,l)\in E,\ s\in S \quad (34) \\ \sum_{i=1}^{n} (d_{si}x_i u_j - d_{si} \sum_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{ijk}) \le e_s(u_j - x_j)\ j\in I,\ s\in S & (35) \end{cases}$$

Here we describe how we get the above valid inequalities (28)-(35):

- Constraints (28) follow from the fact that in any product $x_i x_j$ either x_i or x_j can be replaced by its binary decomposition.
- Constraints (29) follow from the inequality $(x_i - u_i)(x_j - u_j) \geq 0$.
- Constraints (30) define variables z_{iik} that represent $t_{ik}x_i$ for an integer solution.
- Constraints (31) follow from inequality $x_i^2 \geq x_i$ that is satisfied by any integer x_i.
- Constraints (32) are obtained by multiplying the initial equality Constraints (1) by t_{jl}.
- Constraints (33) are obtained by multiplying the initial inequality Constraints (2) by t_{jl}.
- Constraints (34) are obtained by multiplying the initial inequality Constraints (2) by $(1 - t_{jl})$.
- Constraints (35) are obtained by multiplying the initial inequality Constraints (2) by $(u_j - x_j)$.

As in the `BBLr` method, the multiplication of Constraints (1) and (2) by the variables introduces variables z_{ijk} in the new constraints (32)-(35). This is why we need to define Constraints (18')-(27') independently from the sign of q_{ij}. Moreover, variables z_{ijk} become required even when $q_{ij} = 0$.

4 Computational Results

We choose to perform numerical experiments on the Integer Quadratic Knapsack Problem ($IQKP$) that consists in minimizing a quadratic function subject to a linear inequality constraint:

$$(IQKP) \begin{cases} Min\ f(x) = x^T Q x + c^T x & \\ s.t \quad \sum_{i=1}^{n} d_i x_i \leq e & \\ \quad 0 \leq x_i \leq u_i & i \in I \\ \quad x_i \in \mathbb{N} & i \in I \end{cases}$$

We generate instances with 10, 20, and 30 variables. The coefficients are randomly generated as follows:

- the coefficients of Q and c are reals in the interval $[-100, 100]$
- the d_i coefficients are integers in the interval $[1, 50]$
- e is equal to $20 * \sum_{i=1}^{n} d_i$
- we generate a first class of instances, ($IQKP_1$), with all $u_i = 50$, and a second class, ($IQKP_2$), with all $u_i = 100$.

For any size $n = 10$, 20, or 30, we generate 5 instances in each class giving a total of 30 instances.

Our experiments are carried out on a Linux operating system based on an Intel core 2 duo processor, 2.8 GHz with 1024 MB of RAM. We use the modeler and the linear programs solver XPress-Mosel version 1.6.1 (2005) (5).

The results of the four formulations are presented in Tables 1 and 2, where each row corresponds to one instance.

Legenda of the tables:

- n: number of integer variables
- *gap*: $|\frac{b-l}{b}| * 100$ where b is the value of the best known solution and l is the optimal value of the LP relaxation at the root node (in %).
- *nodes*: number of nodes visited by the branch-and-bound algorithm
- *time*: CPU time (in seconds) required by the branch-and-bound algorithm. This time is limited to 1 hour of CPU time.

Program ($LP_{\texttt{BIL}}$) has less variables and constraints than program ($LP_{\texttt{BBL}}$). For example, instances of class $IQKP_1$ with $n = 20$ lead to a program ($LP_{\texttt{BIL}}$) (resp. ($LP_{\texttt{BBL}}$)) with 2820 (resp. 7260) variables and 5061 (resp. 14421) constraints. Moreover, we can observe in Tables 1 and 2 that, for all the instances, the gap associated to ($LP_{\texttt{BIL}}$) is much smaller than the gap associated to ($LP_{\texttt{BBL}}$). Consequently, the BIL approach outperforms the BBL approach with regard to the number of nodes and the computational time.

For BBL and BIL the reinforced versions significantly improve the gap value. Consequently, the number of nodes decreases in these reinforced versions. However, for BBL, the gap improvement is not sufficient to compensate the increase

Table 1. Resolution of ($IQKP_1$) ($u_i = 50$)

	($LP_{\texttt{BBL}}$)			($LP_{\texttt{BBLr}}$)			($LP_{\texttt{BIL}}$)			($LP_{\texttt{BILr}}$)		
n	gap	nodes	time	gap	nodes	time	gap	nodes	time	gap	nodes	time
10	69	1462	51	35	549	88	44	787	9	9	169	15
10	37	478	22	21	423	49	19	449	5	2	13	2
10	59	2316	83	29	577	92	41	886	14	6	66	7
10	41	573	19	31	301	30	19	389	4	0.4	75	5
10	41	403	17	20	129	30	22	319	3	0.3	45	5
20	37	13030	3303	24	863	*(3%)	16	1740	82	0.04	15	22
20	44	10000	*(7%)	26	956	*(7%)	25	3339	169	0.07	5	119
20	55	10000	*(7%)	35	866	*(13%)	33	9322	545	7	95	75
20	45	8218	*(11%)	28	758	*(9%)	23	4355	317	0	1	0
20	38	6435	*(3%)	29	485	*(10%)	23	6323	318	0.6	146	114
30	47	2632	*(36%)	36	270	*(27%)	25	10000	*(8%)	4	148	441
30	79	3288	*(55%)	51	159	*(42%)	52	4813	*(30%)	24	1086	*(11%)
30	45	5833	*(23%)	26	293	*(19%)	22	13739	*(3%)	0.05	81	193
30	84	195	*(60%)	58	171	*(43%)	60	10000	*(40%)	28	1103	*(15%)
30	48	2933	*(6%)	33	175	*(29%)	27	10000	*(11%)	3	568	2088

$*(g\%)$ means that the branch-and-bound is stopped after 1 hour with a MIP gap of $g\%$.

Table 2. Resolution of $(IQKP_2)$ $(u_i = 100)$

	$(LP_{\texttt{BBL}})$			$(LP_{\texttt{BBLr}})$			$(LP_{\texttt{BIL}})$			$(LP_{\texttt{BILr}})$		
n	gap	nodes	time	gap	nodes	time	gap	nodes	time	gap	nodes	time
10	38	503	25	31	143	29	17	423	7	0.1	12	3
10	63	667	44	34	168	52	45	299	7	7	69	5
10	33	362	26	19	206	41	14	162	3	0.1	26	6
10	44	531	24	14	313	50	22	364	4	0.1	87	7
10	37	201	12	5	75	2s	15	251	3	0.1	9	1
20	43	5226	*(18%)	21	900	3524	24	2877	256	0.04	12	28
20	52	4617	996	20	708	*(4%)	29	83	1574	0	1	0
20	63	2900	*(20%)	39	484	*(22%)	38	19528	1130	8	118	123
20	64	6347	*(26%)	38	541	*(22%)	42	16630	1039	11	228	152
20	74	6307	*(22%)	30	385	2848	49	7910	920	4	74	74
30	46	1651	*(37%)	29	48	*(29%)	24	8548	*(1%)	0.4	60	488
30	61	99	*(47%)	23	124	*(29%)	37	1591	*(21%)	3	271	3278
30	44	2219	*(37%)	31	48	*(29%)	22	2956	*(4%)	0.6	255	1828
30	53	414	*(62%)	34	61	*(32%)	31	4452	*(17%)	5	499	3080
30	71	2327	*(39%)	56	108	*(49%)	40	5676	*(9%)	17	824	*(4%)

$*(g\%)$ means that the branch-and-bound is stopped after 1 hour with a MIP gap of $g\%$.

of the size and finally the CPU time required by `BBLr` is larger than the CPU time required by `BBL`.

For `BIL`, the reinforced version leads to an important improvement of the gap, but in this case the improvement of the gap widely compensate the increase of the size and finally the CPU time required by `BILr` is generally significantly smaller than the CPU time required by `BIL`.

We can also observe in Tables 1 and 2 that the gap values associated with `BBLr` and `BIL` are quite similar. However, the size of `BIL` being much lower than that of `BBLr`, `BIL` outperforms `BBLr` from the computational time point of view.

As a conclusion, on these two classes of instances, `BILr` is the best approach for the three criteria : gap, nodes and time. However, the computational experiments have shown that this method was unable to solve instances with 40 variables or more within 1 hour of CPU time.

5 Concluding Remarks

In this paper, we have presented several linear reformulations of linearly constrained quadratic integer programs. The `BBL` and `BBLr` methods that consist in using the standard linearization of quadratic 0-1 programs is not usable because the binary decomposition combined to this linearization leads to 0-1 quadratic programs with too many variables and constraints.

Then, we presented a new approach, `BIL`, using the standard linearization of the product of an integer variable by a binary one. This method reduces significantly the number of constraints and variables added, in comparison with

the BBL approach. In our experiments, surprisingly, this size reduction comes along with a smaller integrality gap. Therefore, BIL is a better approach. Moreover, the valid inequalities added in BILr provide an important improvement. A further improvement would be to incorporate these valid inequalities into a branch-and-cut framework.

References

[1] Fu, H.L., Shiue, C.L., Cheng, X., Du, D.Z., Kim, J.M.: Quadratic Integer Programming with Application in the Chaotic Mappings of Complete Multipartite Graphs. J. Optim. Theory Appl. 110(3), 545–556 (2001)
[2] Fortet, R.: Applications de l'Algèbre de Boole en Recherche Opérationelle. Revue Française De Recherche Opérationelle 4, 17–25 (1960)
[3] Sherali, H.D., Adams, W.P.: A tight linearization and an algorithm for zero-one quadratic programming problems. Management Science 32(10), 1274–1290 (1986)
[4] McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. Mathematical Programming 1(10), 147–175 (1976)
[5] Dash Optimization, Xpress-Mosel version 1.6.1.: Xpress-Mosel language Reference Manual 1.4 (2005), http://www.dashoptimization.com/
[6] Körner, F.: A New Bound for the Quadratic Knapsack Problem and Its Use in a Branch and Bound Algorithm. Optimization 17, 643–648 (1986)
[7] Körner, F.: An efficient branch and bound algorithm to solve the quadratic integer programming problem. Computing 30, 253–260 (1983)

Control of Some Graph Invariants in Dynamic Routing

Mohamed Amine Boutiche

Laboratoire LAID3 USTHB
BP 32 El-Alia 16111 Bab Ezzouar Algiers Algeria
mboutiche@usthb.dz

Abstract. Topology Control is one of principal questions in network design. Tree-decompositions with bags of small diameter models networks, and were used to construct compact routing schemes. Over time, the bags must change to reflect the changes in the network topology as nodes move around, or links failure. It must be possible to restore the service when there is a failure of an edge or a node in the network. In order to preserve the advantages of this structure, we propose to study the case where a node or edge is added to (resp. is removed from) the network and its effects on some invariants of a tree decomposition.

Keywords: Topology Control, Routing, Tree Decomposition, Tree width, Tree length, Graphs.

Introduction

A dynamic network consists of nodes that move freely and communicate with each other using dynamic links. Dynamic networks do not use specialized routers for path discovery and traffic routing. Compact routing scheme consists to support efficient communication between nodes. One way to develop this scheme is to construct tree decomposition architecture; this means that certain nodes must be selected to form the bags. Tree decomposition of minimum tree width were introduce by Robertson and Seymour in 1986 [1]. For these networks with particular topology, Y. Dourisboure [2] showed that it was possible to build a compact routing schemes of deviation to more 2δ (where δ represent a new introduced invariant, called tree-length of a graph) with addresses and local memories of size $O(\delta \log^3 n)$. F.Dragan and I.Lomonosov [3] refine this notion of tree decomposition by introducing acyclic $(R, D)-$clustering, where clusters are subsets of vertices of a graph and R and D are the maximum radius and the maximum diameter of these subsets, the authors achieve a routing scheme of deviation $2R$ with labels of size $O(\log^3 n/\log\log n)$ bits per vertex and $O(1)$ routing protocol for these graphs.

Today, the nature of services and the requests volume in telecommunications industry have radically changes, thanks to the introduction of new technologies which offer large capacities of transmission. Over time, the bags must change to reflect the changes in the network topology as nodes move around, or links

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 52–58, 2008.

failure. Thus, the current networks tend to have an increasingly sparse topology. In this case, the failure of one or more edges (or nodes) can have disastrous consequences if the network does not provide other paths for routing. So, one of the principal questions in the design process of networks is *Topology Control*. Tree decomposition were used too for topology control, that is the problem of determining an appropriate topology for dynamic networks. Readers are referred to Li [4] and Rajaraman [5] for more information on the topic of topology control. It must be possible to restore the service when failure of an edge or a node of the network occurs.

In this work, we study the changes on the invariants "tree-width and tree-length" of the networks which have a topology modeled by a graph which admits a tree decomposition. In order to preserve the advantages of this structure, we propose to study the case where a connection or edge is added to (resp. is removed from) the network and its changes on the two invariants of the tree decomposition. The paper is organized as follows. Section 2 covers some basic definitions on graph theory. Sections 3, 4, 5, 6 presents respectively the study of th cases of addition of an edge, removal of an edge, addition of a vertex and a removal of a vertex. Finally, section 7 concludes with some directions for possible future work.

1 Some Definitions

All graphs occurring in this paper are connected, finite, undirected, without loop and multiple edges. For a subset $S \subset V$ of vertices of G, let $G(S)$ be a sub graph of G induced by S. By $n = |V|$ we denote the number of vertices in G. The distance $dist_G(u, v)$ between vertices u and v of a graph $G = (V, E)$ is the smallest number of edges in a path connecting u and v. The distance between a vertex $u \in V$ and a set S is $dist_G(u, S) = min_{v \in S}\{dist_G(u, v)\}$. The induced diameter is $diam(S) = max_{v,u \in S}\{dist_G(v, u)\}$. We denote by $N_G(v) = \{u \in V : uv \in E\}$ the neighborhood of a vertex v in G and by $N_G[v] = N_G(v)U\{v\}$ the closed neighborhood of v in G. The k-th neighborhood $N_k(v)$ of a vertex v of G is the set of all vertices of distance k to v: $N_kG(v) = \{u \in V : dist_G(u, v) = k\}$.

The notion of tree decomposition was introduced by Robertson and Seymour in their studies of the minors of graphs [1], see figure1 for example.

Definition 1. *A tree decomposition of a graph G is a tree T whose vertices, called bags, are subsets of $V(G)$ such that:*

1. $\cup_{X \in V(T)} X = V(G)$
2. *for all $\{u, v\} \in E(G)$, there exist $X \subset V(T)$ such that $u, v \in X$*
3. *for all $X, Y, Z \subset V(T)$, if Y is on the path from X to Z in T then $X \cap Z \subset Y$*

Tree decomposition is reduced if no bags are contained in another one. A leaf of such decomposition contains necessarily a vertex contained in none other bags. Thus, by induction the tree-length of reduced tree decomposition does not exceed the diameter of a graph.

The width of T is $width(T) = max_{X \in V(T)} |X| - 1$. The length of T is $length(T) = max_{X \in V(T)} diam_G(X)$. The tree width and the tree length of G, denoted by $tw(G)$ and $tl(G)$, are respectively, $min_T width(T)$ and $min_T length(T)$, where the minimum is taken over all tree decomposition of G.

2 Addition of an Edge

The object of this section will be the study of the case where a connection is added to the network, which corresponds to an addition of an edge in the graph G, and its effects on the tree decomposition T of G.

Let $e = uv$ be the added edge to G. Thus, one obtain a new graph $G' = G \cup e$. Now, and starting from T, one wants to obtain an tree decomposition $T'$of G' which is of minimum tree width and tree length. Two cases arise;

2.1 First Case

This case corresponds to that where both extremities u, v of the added edge e are in the same bag in T. Thus, nothing changes for the tree decomposition of G' (ie; G and G' have the same tree decomposition).

$T' = T$

Claim. One has obviously ;$tw(G) = tw(G')$ and $tl(G') \leq tl(G)$.

2.2 Second Case

This case corresponds to that where both extremities u, v of the added edge e do not belong to the same bag in T. Thus, we want to find a tree decomposition T' for G' starting from a tree decomposition T of G which is of minimal tree width and tree length.

T' respects the three rules of definition 1. Therefore, if $e = uv$ is added; One seeks the bags containing u and the bags containing v. Afterwards, one extracts the subtree induced by u from T, denoted T_u , and the subtree induced by v denoted T_v .

A/ $T_u \cap T_v = \emptyset$

Let S be a separator between T_u and T_v, and let suppose that $|T_u| \leq |T_v|$, then one add the vertex v to each bag of T_u and S, in this way, one forces T' to respect rules 2 and 3 of definition 1.

Claim. For T' the tree decomposition of G' obtained starting from G as in $A.1$ or $A.2$, one has ; $tw(G') \leq tw(G) + 1$ and $tl(G') \leq tl(G)$.

Proof. Indeed; if T_u contain at least one bag with $(tw(G) + 1)$ vertices, then the addiction of a vertex v to the bags of T_u imply that T_u will contain at least one bag with $(tw(G) + 2)$ vertices and hence $tw(G') = tw(G) + 1$, otherwise, $tw(G') = tw(G)$. Same reasoninig for T_v .

$tl(G') \leq tl(G)$; it is clear that the addition of an edge makes reduce the distances in the graph $G'$and consequently the length of T', but, one cannot know of exactly how much, in any way, us what interests us, it is that the bound of the tree length of G' is the same one for G. □

Remark 1. One has not the case $T_u \cap T_v \neq \emptyset$, because it will exist a bag $B \in T_u \cap T_v$ such that $B \in T_u$ and $B \in T_v$, hence, one will have $u \in B$ and $v \in B$, then, u and v are in the same bag B in T, contradiction with the hypothesis of second case.

3 Removal of an Edge

The object of this section will be the study of the case where a connection is removed from the network, which corresponds to a removal of an edge in the graph G, and its effects on the tree decomposition T of G.

Let $e = uv$ be the removed edge which is not an isthmus from G. Thus, one obtain a new graph $G' = G \setminus e$, (G' is connective). Now, and starting from T, one wants to obtain a tree decomposition T' of G' which is of minimal tree width and tree length.

If one removes any edge e from G, by definition 1, there exists at least one bag B in T such as both extremities u, v of e belongs to B. Thus, nothing changes for the tree decomposition of G' (ie; G and G' have the same tree decomposition). Thus $T' = T$

Claim. one has ; $tw(G) = tw(G')$ and $tl(G') \geq tl(G)$.

Proof. Since $T' = T$, one has $tw(G) = tw(G')$.

Furthermore, one has $tl(G') \geq tl(G)$; indeed, it is clear that the removal of an edge makes increase the distances in the graph $G'$and consequently the length of T', but, one cannot know of exactly how much., we must compute the length of T', which will constitute the new bound of the tree length of G'. □

4 Addition of a Vertex

The object of this section will be the study of the case where a new user is coming to the network, which corresponds to an addition of a vertex in the graph G, and its effects on the tree decomposition T of G.

Let u be the added vertex to G and $N(x)$ the set of neighbors of x. Thus, one obtain a new graph $G' = (V', E')$ such that $V' = V \cup \{x\}$ and $E' = E \cup \{xy, y \in N(x)\}$. Now, and starting from T, one wants to obtain a tree decomposition T' of G' which is of minimum tree width and tree length. For doing this, we must the following conditions:

1. To respect the rule 1 of the definition, we must place x in at least one bag of T.

2. To respect the second rule of the definition, it must that exists at least one bag containing both x and y, $y \in N(x)$.
3. To respect the third rule of the definition, it must that bags containing x induces a sub tree of T'.

Remark 2. With this method, one will have $|T| = |T'|$, ie; we only place x in one bag of T, we don't create new bags in T'.

For this case, one have $x \in V(G')$, $xy \in E'$, with $y \in N(x)$. Let $y \in N(x)$;One have$x \in V(G)$ and T_y is a sub tree of T induced by bags of T containing y.

A/Case 1: If $\cap_{y \in N(x)} T_y = \emptyset$

Here, with an aim of satisfying the conditions of definition 1, one must place x in bags of T by observing the conditions 2) and 3) referred to above. A trivial solution is to place x in all bags of T, we can do it once, but if we do it twice then the decomposition obtained will not be a tree decomposition and will not respect the rules of definition 1. Hence, we must minimize the number of bags candidate to receive x, ie; minimize the size of T'_x , a sub tree of T' induced by bags containing x in T'. And for this, we applied the following procedure:

Procedure
Input : a tree decomposition Γ'_x of $G' = G \cup \{x\}$ obtained from T by the addition of a vertex x to all bags of T; The set $N(x) =$ Neighbors of x and $\cap_{y \in N(x)} T_y = \emptyset$

Output : the sub tree T'_x of Γ'_x ; induced by the bags containing x verifying conditions 2) and 3) of definition 1, and of minimum cardinality

While $B \in \Gamma'_x$ is a leaf

If $\Gamma'_x \setminus B$ is a sub tree that verify ; It exists at least one bag containing x and y, $y \in N(x)$, then remove B

Else stop;

End

Interpretation

In the tree decomposition Γ'_x, we start by leafs removing such that the remaining sub tree verify the second condition of definition 1. At the end, we obtain a minimal sub tree T'_x that verify conditions 2) and 3), trivially of definition 1. Hence, to obtain a tree decomposition T' of G' from a tree decomposition T of G, for the case **A**, we add the vertex x only to the bags of the sub tree T'_x, that we removed x.

Claim. One has ; $tw(G) \leq tw(G') \leq tw(G) + 1$ and $tl(G') \leq Max(tl(G); LengthT'_x)$, where : T'_x, is a sub tree of T' formed of the bags containing x.

Proof. It's clear that all bags of T containing $(tw(G) + 1)$ vertices are among, those to which, one added the vertex x (to have the tree decomposition T' , hence $tw(G') = tw(G) + 1$, if not then, $tw(G') = tw(G)$.

Bydefinition,onehas:$tl(G') \leq LengthT'$,and$LengthT' = Max_{B \in V(T')} diam_{G'} B$ and $LengthT'_x = Max_{B_x \in V(T')} diam_{G'} B_x$ with B_x bag containing x in T'. As here,

the changes of T' compared to T are the bags B_x, that one determined by adding each time the vertex x and whose sub graph induced by the vertices of $\{Bx, x \in T'_x\}$ contains edges which are in G' and not in G. Thus, to know $LengthT'$, it is enough to know $Max_{B_x \in V(T')} diam_{G'} B_x$ ie; $LengthT'_x$. Thus, if $LengthT'_x \leq tl(G)$ then $LengthT' = tl(G)$ and if $LengthT'_x > tl(G)$ then, one will have $LengthT' = LengthT'_x$, from where ; $LengthT' = Max(tl(G); LengthT'_x)$ and $tl(G') \leq LengthT'$. □

B/Case 2: If $\cap_{y \in N(x)} T_y \neq \emptyset$

Let $B \in \cap_{y \in N(x)} T_y$ with $|B| < |B_i|$, $\forall B_i \in \cap_{y \in N(x)} T_y$. Then, it is enough to add $x \in B$, and one obtains thus, a tree decomposition T' of G' complying with rule 3 of the definition 1.

Claim. One has ; $tw(G') \leq tw(G) + 1$ and $tl(G') \leq Max(tl(G); diam_{G'} B) = LengthT'$.

Proof. It is clear that if $|B| = tw(G) + 1$ in T then the fact of adding a vertex with B implies that $|B| = tw(G) + 2$ in T', thus $tw(G') = tw(G) + 1$. Else $|B| \leq tw(G)$, one will have $tw(G') = tw(G)$, from where $tw(G') \leq tw(G) + 1$.

By definition, one has $tl(G') \leq LengthT'$, and $LengthT' = Max_{B_x \in V(T')} diam_{G'} B$. As here, the only changes of T' compared to T is the bag B to which, one added a vertex and thus the sub graph induced by the vertices of B contain edges which are in G' and not in G. Thus, to know $LengthT'$, it is enough to know $diam_{G'} B$. If $diam_{G'} B \leq tl(G)$, then $LengthT' = tl(G)$. If $diam_{G'} B > tl(G)$, then $LengthT' = diam_{G'} B$, thus $LengthT' = Max(tl(G); diam_{G'} B)$, from where, one has $tl(G') \leq LengthT'$. □

5 Removal of a Vertex

The object of this section will be the study of the case where a user is leaving the network, which corresponds to a removal of a vertex in the graph G, and its effects on the tree decomposition T of G. The case where a vertex is removed from the network corresponds to the case of breakdown of a user.

Let u be this vertex (u not an articulation point), in this case, one will have several connected components.

Thus, $G' = G \setminus u$ is connected.

One wants to determine T' the tree decomposition of G', starting from the tree decomposition T of G.

It is enough to remove u, in the bags of T which contain it, thus, is obtained the tree decomposition T' of $G' = G \setminus u$ and this tree decomposition checks well the three conditions of definition 1.

Remark 3. T' thus obtained will not be inevitably a reduced tree decomposition, and this, same if T is. To return T' a reduced tree decomposition, it is enough to remove the bags which are such that $B \subset B'$ with B and B' adjacent in T.

Claim. One has ; $tw(G) - 1 \leq tw(G') \leq tw(G)$ and $tl(G') \leq LengthT'$.

Proof. Indeed, if all the bags of T which contain $(tw(G)+1)$ vertices are among the bags to which, one removed the vertex u, then ; $tw(G') = tw(G) - 1$. Else $tw(G') = tw(G)$. It is obvious that, $tl(G') \leq LengthT'$ (by definition).

6 Conclusion

In order to study the reliability of a tree decomposition graph model networks, we have considered four cases; addition of an edge, removal of an edge, addition of a vertex and a removal of a vertex, for which, we have to consider a new graph G'. We proved that one can obtain a tree decomposition of minimum tree width and tree length for G' starting from the preceding tree decomposition of G, what returns these models most interesting.

References

[1] Robertsonm, N., Seymour, P.D.: Graph minors.: Algorithmic aspects of tree-width. Journal of Algorithms 7, 309–322 (1986)
[2] Dourisboure, Y., Gavoille, C.: Tree decomposition with bags of small diameter. Discrete Mathematics 307, 2008–2029 (2007)
[3] Feodor, F.: Dragana, Irina Lomonosov; On compact and efficient routing in certain graph classes. Discrete Applied Mathematics 155, 1458–1470 (2007)
[4] Li, X.Y.: Topology control in wireless ad hoc networks. In: Basagni, S., Conti, M., Giordano, S., Stojmenovic, I. (eds.) Ad Hoc Networking. IEEE Press, Los Alamitos (2003)
[5] Rajaraman, R.: Topology control and routing in ad hoc networks: A survey. SIGACT News 33, 60–73 (2002)

A Simulation Tool for Analyzing and Improving the Maternity Block Management

Michelle Chabrol[1], Denis Gallot[2], Michel Gourgand[1], and Sophie Rodier[1,2]

[1] LIMOS UMR CNRS 6158, University of Blaise Pascal, Campus des Cézeaux, 63000 Clermont-Ferrand, France
[2] University Hospital of Clermont-Ferrand, Bd Léon Malfreyt, 63058 Clermont-Ferrand Cedex 1, France
{chabrol,gourgand,rodier}@isima.fr,
{dgallot,srodier}@chu-clermontferrand.fr

Abstract. The pregnant women enter and leave hospitals 24 hours a day throughout the year. So, the designers of models have to consider the day, the time and the method of arrival, the degrees of emergency, the alternative placement, and the staff availability. The discrete event simulation has been widely used in attempts to improve the delivery of healthcare. In this paper, the method used to develop a simulation tool for a maternity block is described. This paper surveys the application of discrete-event simulation modeling to healthcare systems and presents the decision-making aid tool designed for a maternity block in order to improve their management. Future directions of research and applications are also discussed.

Keywords: Modeling, simulation, healthcare, decision-making aid tool, maternity block.

1 Introduction

At the end of 2009, the "Hôtel Dieu", a unit of the University Hospital (UH) of Clermont-Ferrand will transfer its activities on a new site: the "Nouvel Hôpital Estaing" (NHE). The UH managers have to reconsider the organization of the maternity block which will gather in a single place two obstetrics units now distinct: the "Maternity" and the "Polyclinic". The maternity and the polyclinic of the "Hôtel Dieu" are two separated obstetrics units in two different buildings. We define the maternity block as a hospital unit where the pregnant women undergo a medical treatment before and after child birth.

The new obstetrical unit must provide health services for women, which includes the full range of the maternity care and the gynaecology care (only for emergency examinations). The maternity service provides care for pregnant women in the antenatal, intrapartum and postnatal period.

The maternity unit has to respond to an uncertain demand from patients. Some demands can be treated within the maternity block, others are admitted to the hospital for further treatment as inpatients (complications). In this

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 59–68, 2008.

work, our main goals is to provide to the hospital managers and to the medical teams, a decision-making aid tool which allows them to improve and optimize the management of their new structure.

We have built a discrete-event simulation model to show visually and numerically the flow of patients through the maternity block on a typical week. The simulation shows the impact of variability in demand, and process capability.

We start by briefly describing the main simulation approaches used for these types of model, namely discrete-event simulation. We propose our approach for the design of decision-making aid tools, and we present the modeling tools used. Next, we present the decision-making aid tool designed for a maternity block. We conclude with a glimpse at the future.

2 Literature Survey

The literature survey on maternity blocks, except medical publications, is poor. This is partly due to the complexity of this system, which meet very different activities difficult to forecast (the emergency examinations, the delivery, the medical termination of pregnancy, etc.). The maternity block includes different areas, different methods of operation, different asepsis levels and technicality levels (the examination rooms, the operating rooms,. . .), which involve some material and human resources (doctors, midwives, anesthesiologist, operating room nurses, anaesthetic nurses). Unlike a conventional surgical unit, the planned or "programmable" operations do not represent the majority of the activity, and emergencies quickly become vital for the patient, as for baby, requiring management and priority rules very specific. So, it appears necessary to model such systems to develop decision-making aid tools. A choice is to develop a simulation model, that represents all the operations of an unit in sufficient detail for the experimentation of different scenarii of organization and the test of several system loads. Computer or simulation models provide an insight into the working of a system and can be used to predict the outcome of a change in strategy. This is particularly useful when the system is very complex and/or when the experimentation is not possible. Several healthcare administrators have used discrete-event simulation as an effective tool for allocating scarce resources to improve patient flow, while minimising health care delivery costs and increasing patient satisfaction

The potential benefit of the simulation of the health care systems is huge. The literature shows that with no doubt, the most widely used simulation approach in health is discrete-event simulation (DES). In DES, entities have characteristics which determine their pathway through the system, in exactly the same way as patients have individual characteristics which determine their pathway through the hospital system [1]. Model designers have to lead the effort to develop models, by improving their understanding of health care needs and challenges (model designing) and communication of user-friendly models, to help provide answers to the complex health care issues. In this work, Royston [2] studies the future challenges and opportunities in health care modeling and simulation and proposes to compare experiment and modeling (Tab. 1).

Table 1. Experiment vs Modeling

Attribute	Experiment	Modeling and simulation
Veracity	Often high, within its domain	Contingent on logic, elements and data
Timescale	Often long	Can be quick
Cost	Often high	Can be low
Risk	Can be high	Generally low
Extendablity	Limited	High

Jun et al. [3] have surveyed an approximately 30 year period, from the early 1960s to the late 1990s, applications of simulation in healthcare. They have reviewed 117 journal articles and have classified them according to their objectives. Their main interest is the impact of patient and resource scheduling on patient and work flows, followed by the allocation of resources such as beds, rooms and staff. They also have searched for studies of more complex, integrated and multi-facility systems and have concluded that there seems to be a lack of such work reported in the literature. They suggest that the major reasons for this shortage are first, the level of complexity and the data needs. Wilson [4] has surveyed 200 simulation projects in healthcare but has found only 16 projects which reported a successful implementation. Common factors in these 16 projects are at least one author who worked at the institution concerned, a problem of high priority to that institution, an external funding, and/or a detailed description of data collection. Twenty years later, a systematic review of healthcare simulation models [5] has founded 182 papers published between 1980 and 1999. They have identified five broad topic areas: the hospital scheduling and organization, the infection and communicable disease, the costs of illness and the economic evaluation, the screening and the miscellaneous. We are interested in hospital scheduling and organization area. This domain includes 94 (52 per cent) of the 182 papers in the review. Patient scheduling and admissions policies were popular topics for modeling on many sytems : outpatient clinics [6], a walk-in clinic [7], an intensive care unit [8] and operating room scheduling [9] [10] [11]. Fone et al. note that very few papers reported that models had been implemented. The authors said *"... we were unable to reach any conclusions on the value of modeling in health care because the evidence of implementation was so scant."* ([5], p. 333) As Brailsford said [1], one possible barrier to implementation is that of generalizability. All healthcare modelers stress the importance of working closely with clinical or managerial practitioners, in order to gain buy-in and acceptance.

3 Decision-Making Aid Tool Design: Approach and Tools

A simulation model is able to deal with detail complexity by simulating the life histories of individuals and then estimating the population effect from the sum of the individual effects. Each member of the population (entity) included in a simulation model is tracked through a set of options. At each decision point a

variety of choices is available, and the outcome will depend on, for example, the characteristics of the entity and resources, the previous movement through the model, and the choices that other entities have made.

The main dilemma in a such work is deciding on the appropriate level of detail. An increased detail leads to more realistic representation, which should increase the confidence of stakeholders. However, an increased detail requires validated data and it can be expensive and time-consuming to collect all the knowledge of the system.

The abstraction level for the simulation model depend of the modeling level of the formal knowledge. Sinreich and Marmor [12] propose a figure with the range of modeling options and the building blocks used in each case. We adapt and complete it with the classic modeling level for the formalization of the system knowledge (Fig. 1).

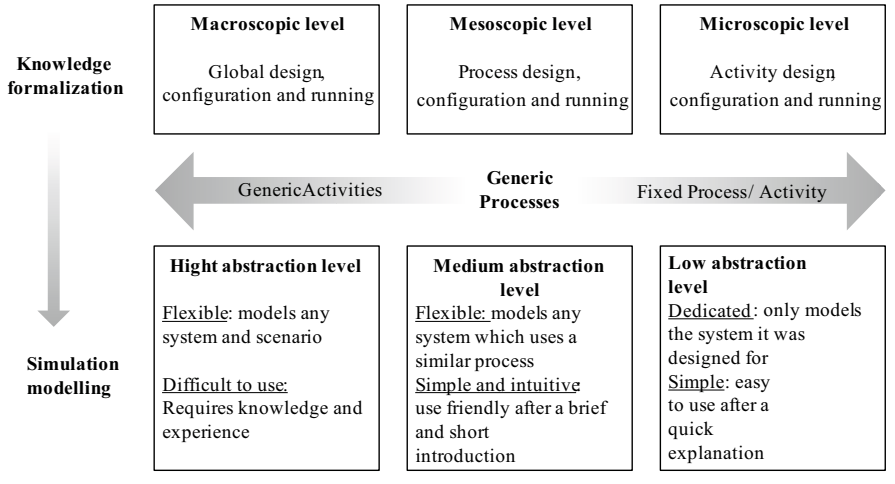


Fig. 1. Abstraction levels for the knowledge formalization and the simulation

The challenge is to develop models at medium abstraction levels, which both consider the system complexity and the ease of use. The main goal of our work is to provide for the maternity block teams, a decision-making aid tool:

- To confirm the physical structure size, to specify the staffing requirements and to plan the resources (functions, allocation...).
- To test and to compare management rules (resources allocation), to study the system response to random events, to test different scenarii (schedules, load, etc.), to improve the service working.
- To estimate the system performances: indicators evaluation, waiting times, occupancy rates, identification of possible bottleneck.

A second goal is to provide coaching for change to the medical teams. Our work should allow the maternity teams and the polyclinic teams to work together on the organisation of the new obstetrical unit. Our approach consists in:

– The data collection on the field (meetings with the executives of health, doctors and midwives).
– The identification of the processes (patient pathways), of the staff tasks , and of the management rules.
– The modeling of the new maternity block.
– The design of a decision-making aid tool based on a simulation model.

We used the ASDI methodology (Analysis, Specification, Design, and Implementation) initially developed by Gourgand [13]. In a previous work [14], we have shown the interest to use this methodology adapted to the hospital systems in order to build, for a given system, knowledge, action and results models which allows to give a decision-making aid and to act on the system. We present the tools used for each stage in the Fig. 2.

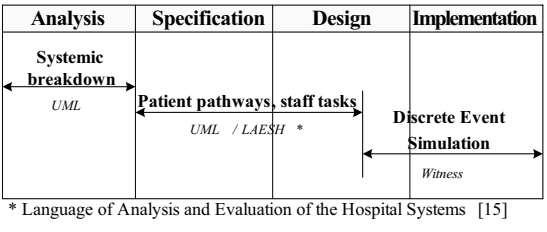


Fig. 2. Approach and tools used

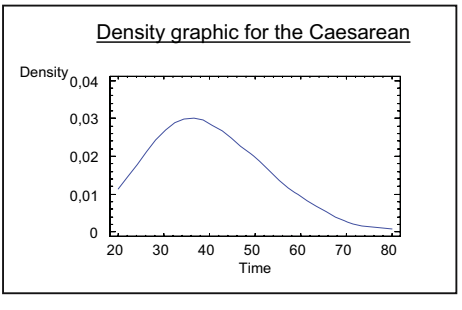


Fig. 3. Density graphic for the Caesarean time

The first step is the design of the knowledge model which formalizes all the entities, the processes and the management rules of the system. We have designed a software component library with Witness software, to satisfy the hospital systems specificities (management and priority rules, preemption,. . .). In a second step, thanks to the knowledge model and to the software component library, we can build an action model, based on a discrete-events simulation model. To feed data into the simulation model, a data analysis was done to determine the different caracteristics for the operation times (distribution laws. . .). Data were obtained from a variety of sources:

centralized information system, local systems, special manual data collection exercises, observation and interview. Fig. 3 shows an example of the statistic work: a density graphic for the time of Caesarean operation. In this paper, we do not present the knowledge model of the maternity block and its translation into simulation model. We just give the different patient pathways idenfied (Tab. 2) and an exemple of knowledge formalization of a patient pathway with LAESH (Fig. 4).

Table 2. Patient pathways

Processes "Father"	
Category 1	Emergency consultation for pregnant women
Category 2	Emergency consultation without pregnancy
Category 3	Childbirth except programmed Caesarean
Category 4	Programmed Caesarean
Category 5	Version by External Operation
Category 6	Medical Interruption of Pregnancy
Processes "Son"	
Category 7	Baby by natural birth
Category 8	Baby by caesarean section

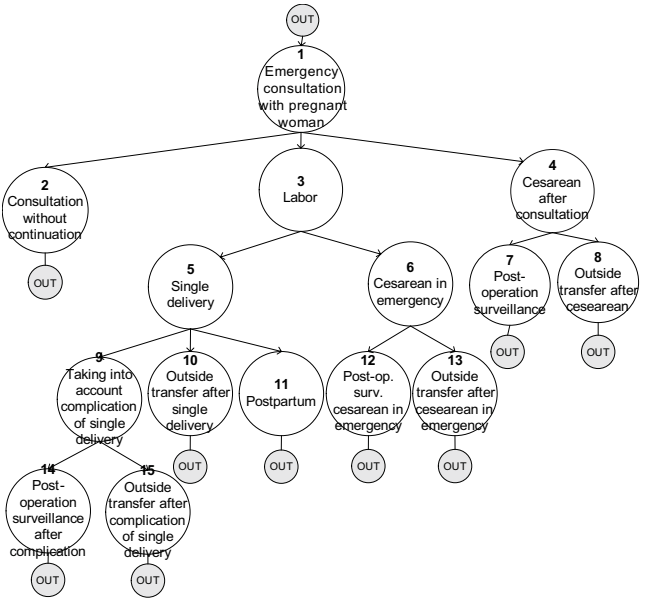


Fig. 4. Example of a patient pathway: Emergency consultation for pregnant women (9 ways)

4 Application: A Decision-Making Aid Tool for a Maternity Block

The primary objective in developing the simulation model is to provide a tool for decision-making in the clinical environment. The simulation tool has to include several modules in order to consider all the requirements described earlier. The tool has to have a Graphical User Interface that is intuitive and simple to use. Through it, the user can input the system characteristics and the other required data and he can get back the different results.

A framework for the development of a generic processes model using the discrete-event simulation and a visually interactive software (Witness) has been created. The framework allows users to input walk-in arrivals hourly, over a 24-hour period, over a 7-day week in the Excel spreadsheet interfaces. It also allows the user to aim the arrivals through a number of different pathways (emergency consultation, programmed Caesarean...) and to set resources within the modeled unit. Input variables of the decision-making aid tool are:

- The human resources with the various time slot of presence (schedules) and the quantity of persons by time schedule/slot;
- The quantity of patients expected by week;
- The patients distribution by "pathway";
- The patients arrival terms by "pathway" (programmed, emergency arrival laws);
- Several probabilities (complications, multiple births, etc.);
- The times of elementary operations (constants, variables).

The interface can generate a patients arrival schedule based on the data and parameters captured (the quantity of patients and the distribution's arrival law). This schedule can be reviewed by the user before running the simulation. The simulation's duration has been set to one week. We have designed a graphical user interface. This graphical interface enables the user to visualize the model. As a communication aid with healthcare professionals, this can be invaluable to validate the model. During the simulation run, the users can see patients arriving in real simulation time, observing monitor their travel through the different work areas as defined by their assigned pathways. The simulation also allows users to see resource use and queue activity throughout the run. A snapshot of the simulation model is shown in Fig. 5. The generic framework also facilitates the testing of new scenarios. The resources can be changed and the outputs can be compared after re-running the model. Similarly, process times can be changed and the model re-run to assess impact. Furthermore, the generic framework could be used to assign other attributes to patients for modeling. The model was used to produce detailed predictions of resource requirements for each scenario.

Two levels of results are obtained. Overall results:

- The overall results: the passive resources (rooms) and active resources occupancy (recorded every fifteen minutes), the total time spent in the system by each patient and the quantity of births of each type (natural childbirth and cesarean section).

– The detailed results which provide: *(i)* For each room; the occupancy times and occupancy rates and the quantity of patients that have been treated; *(ii)* For each type of human resources and each human resources: the times and occupancy rates by place, elementary operation; *(iii)* For each patient: the processing times and the total latency times, by place or elementary operation.

Fig. 6 gives an example of the type of graph than can be obtained with the total occupancy times for each type of human resource in the different areas (consultation, delivery suites, caesarean rooms). Fig. 7 shows the delivery rooms occupancy at intervals of fifteen minutes.

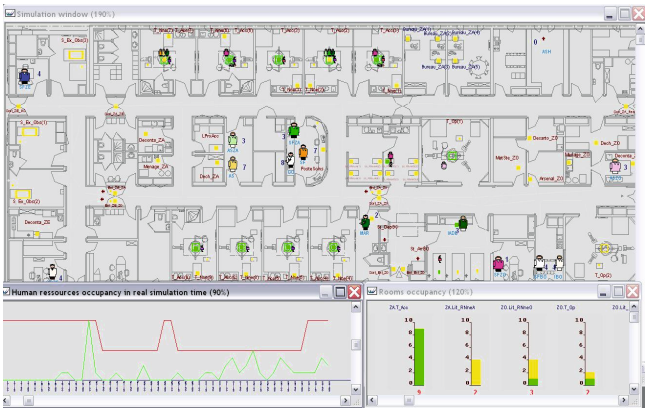


Fig. 5. Snapshot of simulation model

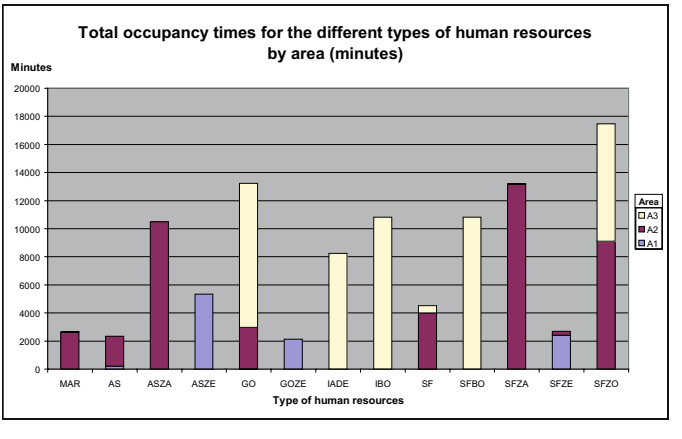


Fig. 6. Human resources occupancy time by area

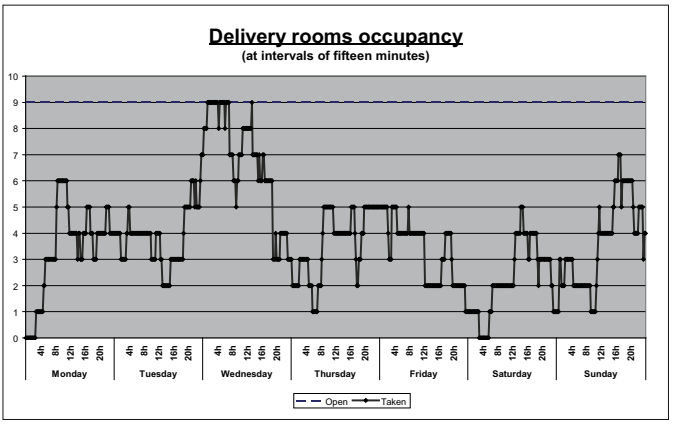


Fig. 7. Delivery rooms occupancy

5 Conclusion

The decision-making aid tool has been developed in close cooperation with staffs in the maternity blocks. It was validated and installed in the maternity blocks of the HU. The medical teams and health care workers staff can test different scenarii from organization (resource allocation, slot time for the programmed interventions...) and adapt their management rules. Moreover, and at the request of physicians and midwives, this tool has been presented at the National Days of the French society for Prenatal Medicine (October 2007), at the Day of Research in Obstetrics and Gynecology (December 2007) and to the 33rd International Conference on Operational Research Applied to Health Services (ORAHS) in July 2007. We now plan to develop couplings between the simulation and optimization approaches.

References

1. Brailsford, S.C.: Tutorial: advances and challenges in healthcare simulation modeling. In: The Winter Simulation Conference, pp. 1436–1448 (2007)
2. Royston See, G.: Trials versus modeling in appraising screening programmes. Br. Med. J. 318, 360–361 (1999)
3. Jun, J.B., Jacobson, S.H., Swisher, J.R.: Applications of discrete event simulation in healthcare clinics: a survey. J. Op. Res. Soc. (JORS) 50, 109–123 (1999)
4. Wilson, J.C.T.: Implementation of computer-simulation projects in healthcare. J. Op. Res. Soc. (JORS) 32, 825–832 (1981)
5. Fone, D., Hollinghurst, S., Temple, M., Round, A., Lester, N., Weightman, A., Roberts, K., Coyle, E., Bevan, G., Palmer, S.: Systematic review of the use and value of computer simulation modeling in population health and health care delivery. J. Pub. H. Med. 25, 325–335 (2003)
6. Stafford Jr., E.F., Aggarwal, S.C.: Managerial analysis and decision-making in outpatient health clinics. J. Op. Res. Soc. (JORS) 30, 905–915 (1979)

7. Reilly, T.A., Marathe, V.P., Fries, B.E.: A delay-scheduling model for patients using a walk-in clinic. J. Med. Syst. 2, 303–313 (1978)
8. Kim, S.-C., Horowitz, I., Young, K.K., Buckley, T.A.: Analysis of capacity management of the intensive care unit in a hospital. Eur. J. Op. Res. (EJOR) 115, 36–46 (1999)
9. Dexter, F., Macario, A., Traub, R.D., Hopwood, M., Lubarsky, D.A.: An operating room scheduling strategy to maximize the use of operating room block time: computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time. Anesthesia and analgesia 89, 7–20 (1999)
10. Fitzpatrick, K.E., Baker, J.R., Dave, D.S.: An application of computer simulation to improve scheduling of hospital operating room facilities in the United States. Int. J. Comp. App. Teh. 6, 214–224 (1993)
11. Ballard, S.M., Kuhl, M.E.: The use of simulation to determine maximum capacity in the surgical suite operating room. In: The Winter Simulation Conference, pp. 433–438 (2006)
12. Sinreich, Marmor: A simple and intuitive simulation tool for analyzing emergency department operations. In: The Winter Simulation Conference, pp. 1994–2002 (2004)
13. Gourgand, M.: Outils logiciels pour l'évaluation des performances des systémes informatiques. Doctorat d'Etat, University of Blaise Pascal, Clermont-Ferrand, France (1984)
14. Chabrol, M., Féniès, P., Gourgand, M., Tchernev, N.: Un environnement de modélisation pour la Supply Chain Hospitaliére: application sur le. Nouvel Hôpital d'Estaing. Ingénierie des Systémes d'Information 11, 137–162 (2006)
15. Chabrol, M., Gourgand, M., Rodier, S.: A modeling methodology and its application to the design of decision-making aid tools for the hospital systems. In: IEEE International Conference on Research Challenges in Information Science (2008)

Solving the Multiple Objective Integer Linear Programming Problem

Mohamed El-Amine Chergui[1], Mustapha Moulaï[1], and Fatma Zohra Ouaïl[2]

[1] Laboratory LAID3, Faculty of Mathematics, USTHB BP 32, El Alia 16111, Algeria
mchergui@usthb.dz
[2] Département de Recherche Opérationnelle, Faculté des Mathématiques
USTHB BP. 32, EL ALIA 16111 Alger, Algérie

Abstract. In this work, an exact method for generating the efficient set of the multiple objective integer linear programming problem (MOILP) is described. When many of the published methods consist of solving initially an ILP program, our method has the advantage of starting with an optimal solution of an LP program whose objective is a positive combination of the criteria, and uses a branching procedure to generate an integer feasible solution. Whenever such a solution is found, the increasing directions of the criteria are recognized and an efficient cutting plane is built in order to delete some of the non efficient solutions without computing them. Compared to the Sylva & Crema's method where at each stage, the ILP programs considered are augmented by $(q+1)$ new constraints and q bivalent variables, our method does not depend on q, where q is the number of the criteria.

1 Introduction

Multiple objective integer linear programs (MOILP) are often adequate models for many real-world situations. With such a formulation, an important point is to be able to generate the set of all efficient solutions to this problem (see for example [2]).This paper proposes a novel algorithm to do so. The main originality of the approach taken in the present paper is to make use of classical branching well known in the branch and bound technique (see for example [6]) with a novel efficient cut. Several methods have been developed to generate all efficient solutions of the MOILP problem ([1], [3], [6], [9], [10], [11]). For instance, Klein & Hannan [5], gave an implicit enumeration algorithm which consists of solving a sequence of single objective integer linear programs progressively more constrained. The additional constraints exclude both previously generated efficient solutions and some of the non efficient ones. In [8], a variation of the [5] algorithm is proposed, maximizing at each step a positive combination of the q objective functions ensuring the detection of an efficient solution. The ILP programs considered at each stage are augmented by $(q+1)$ new constraints and q bivalent variables which aims to eliminate some of the non efficient solutions. The number of ILP problems to be solved is given by the number of nondominated solutions plus one corresponding to an unfeasible problem. The method

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 69–76, 2008.

proposed by Gupta & Malhotra [4], is also a variant of that proposed by Klein & Hannan [5]. It is to determine the set of all efficient solutions where the authors were able to reduce the number of additional constraints at each stage of the procedure. Unfortunately, an example showing that the algorithm stops before producing all efficient solutions is constructed. Therefore, the approach adopted by the authors does not always provide the entire set of efficient solutions. A synthesis of research tasks is made by [10] and completed by [3].

In this paper, a new exact approach based on a branch and cut technique is developed to generate all efficient solutions of the MOILP problem, without computing all feasible integer solutions. In the proposed method, we do not need to search for an initial optimal solution for an ILP problem. Indeed, based on a simplex method, the criteria evolve in a dynamic way in an augmented simplex table when a linear programming problem is solved, then a branching process is carried out to detect an integer solution for a constrained problem. When such a solution is obtained, it is tested for efficiency with those already found using the criteria values. The increasing directions of the criteria are used to build efficient cuts, in order to avoid the exploration of domains containing integer solutions but not efficient integer ones.

The sections described in this paper are organized as follows: after a formal introduction of the problem in section 2, the principle of the method is reported in section 3. The main theoretical results allowing to justify various stages of the algorithm suggested in the preceding section are developed in section 4. Computational results are reported in section 5 and a final section concludes.

2 Problem Formulation

We consider the following *multiple objective integer linear programming problem* (MOILP):

$$(P)\begin{cases} \max\ Z^1 = c^1 x \\ \max\ Z^2 = c^2 x \\ \quad\vdots \\ \max\ Z^r = c^r x \\ x \in S \\ x\ integer \end{cases}$$

where $S = \{x \in R^n | Ax = b, x \geq 0\}$ is the feasible set and $Z^q = c^q x,\ q = 1, \ldots, r$, $r \geq 2$, are real-valued linear functions, $c^q = \left(c_j^q\right)_{j=1,\ldots,n}$. We assume that S is a nonempty, compact polyhedron, all components of the $m \times n$ matrix $A = (a_{ij})_{i=1,\ldots,m;\ \ j=1,\ldots,n}$ and the m vector b are integers.

A solution x is known as *efficient solution*, if there is not another solution y such that $c^q y \geq c^q x$ for all $q \in \{1, \ldots, r\}$ and $c^q y > c^q x$ for at least one index $q \in \{1, \ldots, r\}$. Otherwise, x is not efficient and the vector Cy dominates the vector Cx, where $C = (c^q)_{q \in \{1,\ldots,r\}}$.

An ideal point x is a solution that maximizes all criteria at the same time: $Cy \leq Cx$ for any feasible solution y.

We define the linear program (P_l) as follows:

$$(P_l)\begin{cases}\max Z = \sum_{q=1}^{r} \lambda_q c^q x \\ x \in S_l\end{cases}$$

with $\lambda_q \geq 0\ \forall q = 1, ..., r\ ;\ S_0 = S$.

At each stage that an integer solution x_l^* is obtained after the branching process, the following definitions and notations are used:
B_l and N_l are respectively, the indices sets of basic variables and non-basic variables of x_l^*, $H_l = \left\{j \in N_l | \exists q \in \{1, ..., r\}\,; \widehat{c}_j^q > 0\right\} \cup \left\{j \in N_l | \widehat{c}_j^q = 0, \forall q \in \{1, ..., r\}\right\}$, where $\widehat{c}_j^q$ is the j^{th} component of the reduced cost vector of the criterion Z^q. The two following subsets of set S_l are defined:

$$S_{l+1} = \left\{x \in S_l | \sum_{j \in H_l} x_j \geq 1\right\} \text{ and } T_{l+1} = \left\{x \in S_l | \sum_{j \in N_l \setminus H_l} x_j \geq 1\right\}.$$

3 Principle of the Method

The method is based on the concept of branching in integer linear programming. All operations described below are identified in nodes and branches in a structured tree. At each node, we have to solve a program (P_l). A node l of the tree is saturated if the corresponding program (P_l) is not feasible or if $H_l = \emptyset$. If the optimal solution x of program (P_l) is not integer, let x_j be one coordinate such that $x_j = \alpha_j$ where α_j is a fractional number. Then, the node l is separated in two nodes with the additional constraint respectively: $x_j \leq \lfloor \alpha_j \rfloor$ and $x_j \geq \lfloor \alpha_j \rfloor + 1$, where $\lfloor \alpha_j \rfloor$ indicate the greatest integer less than α_j. Each corresponding branch define a new linear program (P_k), $k > l$, to solve.

The case corresponding to an integer solution x is solved by using the increasing directions of criteria to avoid exploring non efficient regions of the feasible solutions of problem (P). Only the part of feasible solutions domain in which at least one of the objectives of problem (P) can be improved is treated. This is made possible by adding the following valid constraint that we call the efficient cut: $\sum_{j \in H_l} x_j \geq 1$.

Algorithm

Step 1. (*Initialization*)
$S_0 := S$, $l := 0$ and $Eff := \emptyset$; (integer efficient set of problem (P))
Let $\lambda_q \geq 0$ for all $q \in \{1, \ldots, r\}$, with at least one strict inequality, solve the linear program (P_0) at node 0 and let x be an optimal solution.
If x is not integer, go to *Step 2a*, else go to *Step 2b*.

Step 2. (*General Step*)
As long as there is an unsaturated node in the tree, do:
Choose the first created node l of the tree, not yet saturated and solve the

corresponding linear program (P_l). If program (P_l) have not feasible solutions, then the corresponding node l is saturated. Else, let x be an optimal solution. If x is not integer, go to *Step 2a*. Else, go to *Step 2b*.

Step 2a. Choose one coordinate x_j of x such that $x_j := \alpha_j$, with α_j fractional number, and separate the actual node l of the tree in two nodes: add the constraint $x_j \leq \lfloor \alpha_j \rfloor$ in the first node, the constraint $x_j \geq \lfloor \alpha_j \rfloor + 1$ in the second node and go to *Step 2*.

Step 2b. If Cx is not dominated by Cy, for all solution $y \in Eff$, then $Eff := Eff \cup \{x\}$. If there exists $y \in Eff$ such that Cy is dominated by Cx, then $Eff := Eff \backslash \{y\} \cup \{x\}$. Determine the sets B_l, N_l and H_l, If $H_l = \emptyset$, then the corresponding node l is saturated, go to *Step 2*. Else, add the constraint $\sum_{j \in H_l} x_j \geq 1$ to obtain the set S_{l+1}, solve the corresponding program using the dual simplex method and let x be an optimal solution, If x is an integer solution, go to *Step 2b*. Else, go to *Step 2a*.

4 Main Result

In this section, justifications of steps described in the above method are established. The following results show that, at each step l of the method, no integer efficient solution of the set S_l can be ignored when we consider the set $S_{l+1} \subset S_l$. Consider the optimal simplex tableau with the integer solution x_l^* and note $D' = \{x \in D | x \text{ integer and } x \neq x_l^*\}$, $D \subseteq S_l$.

Lemma 1. $S'_l = S'_{l+1} \cup T'_{l+1}$

Proof. Let $x \in S'_l$. Then x is in the closed domain generated by the Dantzig cut $\sum_{j \in N_l} x_j \geq 1$. As H_l and $N_l \backslash H_l$ define a partition of set N_l, the Dantzig cut can be written as $\sum_{j \in H_l} x_j + \sum_{j \in N_l \backslash H_l} x_j \geq 1$.

If the solution x satisfies the inequality $\sum_{j \in H_l} x_j \geq 1$, then $x \in S'_{l+1}$. If not, $\sum_{j \in N_l \backslash H_l} x_j \geq 1$ and hence $x \in T'_{l+1}$.

Consequently, $x \in S'_{l+1} \cup T'_{l+1}$, and $S'_l \subseteq S'_{l+1} \cup T'_{l+1}$.

In the other hand, it is clear that $S'_{l+1} \cup T'_{l+1} \subseteq S'_l$ and we can conclude that the equality is true. □

Theorem 1. *Let $x \neq x_l^*$ be an integer efficient solution in domain S_l, then x is located in set S'_{l+1}.*

Proof. Let $x \neq x_l^*$ be an integer solution in domain S_l such that $x \notin S'_{l+1}$, then by the above lemma $x \in T'_{l+1}$. Hence, the coordinates of x check the following inequalities: $\sum_{j \in H_l} x_j < 1$ and $\sum_{j \in N_l \backslash H_l} x_j \geq 1$.

This is equivalent to the following conditions: $x_j = 0$ for all $j \in H_l$, and $x_j \geq 1$ for at least one index $j \in N_l \backslash H_l$. By using the simplex table in x_l^*, the following equality holds for all criterion $q \in \{1, ..., r\}$:

$$
\begin{aligned}
& c^q x = c^q x_l^* + \sum_{j \in N_l} \widehat{c}_j^q x_j \\
\Longrightarrow\ & c^q x = c^q x_l^* + \sum_{j \in H_l} \widehat{c}_j^q x_j + \sum_{j \in N_l \backslash H_l} \widehat{c}_j^q x_j \\
\Longrightarrow\ & c^q x = c^q x_l^* + \sum_{j \in N_l \backslash H_l} \widehat{c}_j^q x_j
\end{aligned}
$$

Hence $c^q x \leq c^q x_l^*$ for all criterion $q \in \{1, ..., r\}$, with $c^q x < c^q x_l^*$ for at least one criterion since $\widehat{c}_j^q \leq 0$ for all $j \in N_l \backslash H_l$.

We conclude that solution x is not efficient and then, all efficient integer solutions in domain S_l belong to domain S'_{l+1}. □

Corollary 1. *The constraint* $\sum_{j \in H_l} x_j \geq 1$ *define an efficient cut.*

Proof. It is clear that $\sum_{j \in H_l} x_j \geq 1$ is an efficient valid constraint by the above theorem, since all integer efficient solutions in the current domain S_l check this constraint. Moreover, the current integer solution x_l^* does not satisfy this constraint since $x_j = 0$ for all $j \in H_l$. In conclusion, we can say that the constraint $\sum_{j \in H_l} x_j \geq 1$ is an efficient cut. □

Theorem 2. *The algorithm described bellow generates all the efficient integer solutions and terminates in a finite number of iterations.*

Proof. The set S of feasible solutions of problem (P) being compact, it contains a finite number of integer solutions. At each step l of the algorithm, one determines an integer solution x_l^* when there exists. By taking into account the lemma and theorem above, at least the solution x_l^* is eliminated when a cut is added. In the other hand, when the set H_l is empty the corresponding solution x_l^* is an ideal point and the current node can be saturated since no criterion can be improved. □

5 Computational Results

The method was implemented in a Matlab 7.0 program, using PC pentium 4, CPU 1.60 GHz 512 MB RAM. All the procedures in the method were programmed by our students and no packages are used. The method was tested with randomly generated m constraints, $m \in \{5, 10\}$ and r objective functions, $r \in \{4, 10\}$. The coefficients are uncorrelated integers uniformly distributed in the interval $[20, 90]$ for constraints and $[30, 100]$ for objective functions. For each constraint j, the right-hand side value b_j is set to maximum value between $\alpha\%$ of the sum of the coefficients (integer part) of each constraint, where $\alpha \in \{17, 20, 25\}$, and the maximum value of the coefficients. Problems with n variables, $n \in \{20, 25, 30\}$, are considered. The variables are bounded and take the possible values $0, 1$ and 2. For each instance (n, m, r, α), a sequence of 20 problems is solved and the whole efficient solution set was generated for all these problems. In the last column of each table, EC/C indicate the ratio between the number of efficient cuts and the total number of the used cuts.

Table 1. Computational results

(n, m, r, α)	Efficient Solutions		CPU(s)		Efficient Cuts		EC/C
	Mean	Max	Mean	Max	Mean	Max	Mean
(20,5,4,25)	99.7	216	281.6	342.2	241.6	340	96%
(20,10,4,25)	60.1	228	265.8	306.9	157	195	98.9%
(20,10,10,25)	763.6	1889	311.73	378.45	147.3	182	99%
(25,5,4,20)	76.5	152	1280.1	1609.2	665.5	807	99.8%
(25,10,4,20)	48.8	140	1082.5	1179.4	365,3	434	99.7%
(25,10,10,20)	1277.4	3749	1354.6	1853.5	378.7	438	98%
(30,5,4,17)	119.9	300	2224.4	2553	905.4	1019	99.8%
(30,10,4,17)	36.4	47	2167.7	2351.4	574.2	624	99.8%
(20,20,4,25)	24.35	48	46.92	67.7	382.6	588	100%
(25,25,4,20)	35.8	142	184.29	255.4	771.1	1242	100%
(30,30,4,17)	33.2	119	587.3	870.6	963.4	2062	100%

In the first experimentation, the MOILP problems considered are of general form and the cumulative results are reported in the first part of Table 1. Obviously in this case, the results show that the CPU time increases rapidly with the data size, the method being exact, but the number of criteria does not significantly increase the CPU time. In the other hand, it should be noted that the ratio EC/C tends towards 100%, which proves that almost all the added cuts are efficient cuts. This means that in most cases, the built set H_l is different from the set N_l of non basic indices variables. Let us note also that the method becomes faster when the size of the set H_l is small, because the domain to remove from the feasible solutions set is more large. This happens particularly when the constraints matrix is triangular. In the second part of Table 1, we have reported the results of particular MOILP treated problems with triangular constrained matrices. In this case, the CPU time decreased, but in the same time, the number of efficient cuts increased. This proves that the removed domains contain many non efficient integer solutions that the method will not have to generate.

The method described in [8] was also programmed for searching the integer efficient set. We give hereafter the results obtained:

Table 2. Comparative results given on average with 4 objectives

n	m	Eff	Our method		Sylva & Crema	
			iter	CPU(s)	iter	CPU(s)
5	5	21.2	577	0.34	51569	813.68
10	5	25.2	1314.6	0.59	142111.6	3058.37
15	5	24.8	2720.8	1.27	187036	2830.17
10	10	20.6	1018.8	0.64	66245.6	907.89
15	10	16.8	2086	0.82	51976	291.94

where iter represents the number of simplex iterations and Eff is the cardinality of the efficient set.

As expected, the size of ILP problems to be solved is closely related to the number of criteria and the CPU time grows faster with the size of the data compared to our method. However, a subset of efficient solutions can be obtained as soon as the calculation is interrupted unlike our method which gives the efficient set at the end of the algorithm.

6 Conclusion

In this paper, a new exact method combining the well-known principle of branching in integer linear programming with a new efficient cut is described to generate all integer efficient solutions of a MOILP problem. It can be considered as a general method dedicated to MOLP problems with integer as well as zero-one decision variables can be solved by the method. The comparative study proves that our method is faster than that proposed by Sylva & Crema whose CPU time increases quickly of one iteration to another, because of the additional constraints and variables. The method was tested only on medium size problems since the problem is too difficult to solve. However, the tree structure of the proposed algorithm can be parallelized in order to allow the resolution of large size problems.

Acknowledgement

The authors are grateful to anonymous referees for their substantive comments that improved the content and presentation of the paper.

This work was supported by the Laboratory LAID3, USTHB.

References

1. Abbas, M., Moulaï, M.: Solving multiple objective integer linear programming. Ricerca Operativa 29/89, 15–38 (1999)
2. Alves, M., Clìmaco, J.: A Review of Interactive Methods for Multiobjective Integer and Mixed-Integer Programming. EJOR 180, 99–115 (2007)
3. Clìmaco, J., Ferreira, C., Captivo, M.: Multicriteria integer programming: An overview of the different algorithmic approaches. Multicriteria Analysis 2, 248–258 (1997)
4. Gupta, R., Malhotra, R.: Multi-criteria integer linear programming problem. Cahiers de CERO 34, 51–68 (1992)
5. Klein, D., Hannan, E.: An algorithm for multiple objective integer linear programming problem. EJOR 9, 378–385 (1982)
6. Korhonen, P., Wallenius, J., Zionts, S.: Solving Discrete Multiple Criteria Problem Using Convex Cones. Management Science 3011, 1336–1345 (1984)
7. Nemhauser, G.L., Wolsey, L.A.: Integer and combinatorial optimization. John Wiley & Sons, New York (1988)

8. Sylva, J., Crema, A.: A method for finding the set of non-dominated vectors for multiple objective integer linear programs. EJOR 158(1), 46–55 (2004)
9. Steuer, R.E.: Multiple Criteria Optimization: Theory,Computation and Applications. John Wiley & Sons, New York (1985)
10. Teghem, J., Kunsch, P.: A survey of techniques to determine the efficient solutions to multi-objective integer linear programming. Asia Pacific Journal of Oper. Res. 3, 95–108 (1986)
11. Ulungu, E.L., Teghem, J.: Multi-objective Combinatorial Optimization Problem: A Survey. Journal of Multi-Criteria Decision Analysis 3, 83–104 (1994)

Generalized Polychotomic Encoding: A Very Short Bit-Vector Encoding of Tree Hierarchies

P. Colomb[1], O. Raynaud[1], and E. Thierry[2]

[1] LIMOS, Blaise Pascal University, Clermont-Ferrand
{colomb,raynaud}@isima.fr
[2] LIAFA, Univ. Paris 7 & Univ. de Lyon, ENS Lyon
ethierry@ens-lyon.fr

Abstract. A well-known method to represent a partially ordered set P consists in associating to each element of P a subset of a fixed set $S = \{1, \ldots, k\}$ such that the order relation coincides with subset inclusion. Such an embedding is called a *bit-vector encoding* of P. Such encodings are economical with space and comparisons between elements can be performed efficiently via subset inclusion tests. As a consequence, they have found applications in databases, knowledge representation, distributed computing or object-oriented programming. The main issue consists in minimizing the *size* of the encoding, i.e. the cardinal of S, in order to get the best storage space and comparison speed. This smallest size is called the *2-dimension* of P. Its computation is known to be $\mathcal{NP}$-hard in the general case [1] and the complexity is open for trees which are an important class of orders encountered in practice.

Finding heuristics which provide encodings of small size is challenging and it has yielded many works in the general case and in the particular case of trees. Our paper presents a new algorithm for trees which improves all previously known heuristics for trees.

1 Introduction

Partially ordered sets (*orders* for short) occur in numerous fields of computer science, like distributed computing, programming languages, databases or knowledge representation. Such applications have raised the need for storing and handling them efficiently. Many ways of encoding partially ordered sets have been proposed in the literature. Depending on the purposes, several criteria are commonly considered to guide the choice of the most appropriate encoding. One may cite the compromise between speeding up operations and saving space, the choice between dynamic or static data structures with regard to possible modifications of the order, the complexity of generating the encoding from usual data structures (like matrices or lists of successors), the restrictions on the data structures imposed by hardware and software (e.g. storing the order in a database which can be then accessed only by means of SQL requests). Performing fast comparisons between elements while saving space is the most usual issue.

Here is a non-exhaustive list of approaches that have been studied: numbering the elements in order to compress their lists of successors [2,3], partitioning

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 77–86, 2008.

the order into nice subsets like antichains [4,5,6] or chains [7,5,8,9,10], mixing numbering and partitioning [11,12], seeing the order as the inclusion order on some geometrical shapes [13,14], describing the order as the union of nice orders on the same set of elements [15,16], describing the order by combinations of boolean formulas on integer tuples [17,18,19,20], focusing on lattice operations [21,22], embedding the order into another one which is known to have a nice representation [23,24].

In this article, we study *bit-vector encodings* of orders which are embeddings into boolean lattices. In other words, let $P = (X, \leq_P)$ be an order, a *bit-vector encoding* of P is a mapping ϕ from X into 2^S (the set of all the subsets of a set S, ordered by inclusion) such that for all $x, y \in X$, $x \leq_P y$ if and only if $\phi(x) \subseteq \phi(y)$. The *size* of the encoding ϕ is the cardinal of S. It is well-known that there always exists a *canonical* bit-vector encoding embedding P into 2^X and defined for all $x \in X$ by $\phi(x) = \{y \in X | y \leq_P x\}$. A classical implementation of *bit-vector encodings* associates to each element x a vector V_x of $|S|$ bits where bit i is equal to 1 if $i \in \phi(x)$ and equal to 0 otherwise. In that case, checking whether $x \leq_P y$ is equivalent to check whether V_x OR $V_y = V_y$ on the vectors. Fig. 1 illustrates the two representations of such embeddings.

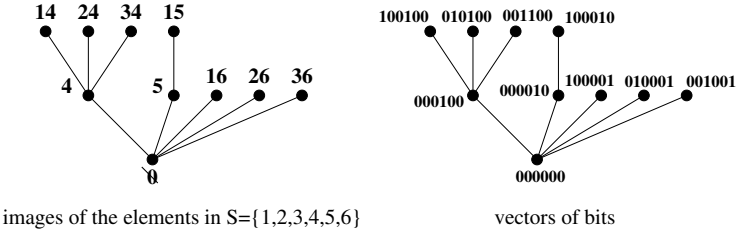


Fig. 1. The two representations of a bit-vector encoding.

Bit-vector encodings provide a compact way to store an order. The size of a bit-vector encoding can be really lower than the n bits per element required for instance by the binary matrix storage of the order relation (where n is the number of elements). Concerning the speed of the inclusion tests, checking whether V_x OR $V_y = V_y$ on the vectors of bits uses elementary bitwise boolean operations, and the speed is proportional to the size of the encoding divided by the length of machine words. For those reasons, such encodings have been used for several types of applications, *e.g.,* databases [21], knowledge representation [25], object oriented programming [6].

The critical parameter is the size of a bit-vector encoding and its minimization improves both space compression and comparison speed. Given an order P, the smallest size of a bit-vector encoding of P is called the *2-dimension* of P and denoted $\mathbf{Dim_2}(P)$. Originally defined in 1963 [26], this parameter has yielded many studies in mathematics and later in computer science. Its computation is known to be $\mathcal{NP}$-hard in the general case (see [1] for a survey). Nevertheless the assets of bit-vector encodings have urged to design good heuristics for

applications. Beyond algorithms for the general case [], the class of trees has been specifically studied by several authors: it belongs to many classical classes of orders and in many applications, the orders involved are trees. Note that since we deal with orders, those trees are rooted.

Our contribution is the design of a new heuristic for trees called *Generalized Polychotomic Encoding* providing bit-vector encodings of very small size. Although it does not compute the *2-dimension*, it improves the best known heuristic for trees designed by Filman and called *Polychotomic Encoding* [27]. Section 2 surveys previous results about bit-vector encodings of trees and presents a general formulation of several heuristics from literature. Using this formulation, we show in Section 3 how to improve Filman's heuristic. We first prove that for any tree, the size of the encoding produced by our algorithm is always smaller than Filman's one. Then in Section 4, we apply our algorithm to a set of benchmarks to compare our algorithm with *Polychotomic Encodings* and *Dichotomic Encodings*, another former heuristic.

2 State of the Art

2.1 Previous Works

The 2-dimension of two particular cases of trees is known for long: *chains* and *antichains*. A *chain* $P = (X, \leq_P)$ is an order such that $\forall x, y \in X$, $x \leq_P y$ or $y \leq_P x$. An *antichain* $P = (X, \leq_P)$ is an order such that $\forall x, y \in X$, neither $x \leq_P y$ nor $y \leq_P x$.

Proposition 1 (Folklore). *Given a chain $P = (X, \leq_P)$ with n elements, then $\mathbf{Dim_2}(P) = n-1$. Let $x_0, x_1, \ldots, x_{n-1}$ be the n elements of P ordered by $x_0 <_P x_1 <_P \ldots <_P x_{n-1}$, then an optimal bit-vector encoding ϕ using colors from $S = \{1, \ldots, n-1\}$ is given by $\phi(x_0) = \emptyset$ and $\phi(x_i) = \{1, \ldots, i\}$ for all $1 \leq i \leq n-1$.*

Proposition 2 (Sperner [28]). *Given an antichain $P = (X, \leq_P)$ with n elements, then $\mathbf{Dim_2}(P) = sp(n)$ where $sp(n) = \min\{k | \binom{k}{\lfloor k/2 \rfloor} \geq n\}$. An optimal bit-vector of P is obtained by associating with each $x \in X$ a combination of $\lfloor sp(n)/2 \rfloor$ elements from $S = \{1, \ldots, sp(n)\}$.*

There exists a tight approximation of the numbers $sp(n)$ for $n \geq 2$.

Proposition 3 ([1]). *Let $n \geq 2$ and $sp(n) = \min\{k | \binom{k}{\lfloor k/2 \rfloor} \geq n\}$. Then*

$$\log_2(n) + \log_2 \log_2(n)/2 < sp(n) < \log_2(n) + \log_2 \log_2(n)/2 + 2.$$

Hence $sp(n) \in \{\lfloor \log_2(n) + \log_2 \log_2(n)/2 + 1 \rfloor, \lfloor \log_2(n) + \log_2 \log_2(n)/2 + 2 \rfloor\}$.

For instance, $sp(2) = 2$, $sp(3) = 3$, $sp(4) = 4$, $sp(5) = 4$ and $sp(100000) = 20$.

Caseau is the first to have studied bit-vector encodings for the whole class of trees [29]. Let C be a chain $x_0, x_1, \ldots, x_p$ of a tree T, he defines the *weight* of this chain by $weight(C) = \sum_{0 \leq i \leq p} deg(x_i)$ (where $deg(x_i)$ is the number of

children of x_i). The size of Caseau's encoding for T is the maximum weight over all the chains of T. After a first improvement by Krall, Vitek and Horspool [30] based on the coloring of a conflict graph, Caseau, Habib, Nourine and Raynaud provided a better heuristic where the output size for a tree T is the maximum over all the chains C of $weight(C) = \sum_{0 \leq i \leq p} sp(deg(x_i))$ [31]. Raynaud and Thierry then introduced a new heuristic based on a balancing principle [32]. Their *Dichotomic Encodings* were improved by Filman's *Polychotomic Encodings* [27] who managed to generate bit-vector encodings smaller than all previously known heuristics.

2.2 A Generic Algorithm

The following algorithm can be viewed as a generic method to encode trees. It uses a depth-first search of the original tree to determine the number of bits needed for distinguished each children of each node. For a node t with children $(x_1, ..., x_k)$ the algorithm computes recursively the number of bits needed to represent each node x_i (so called *weight*). Nodes with no children have a weight of 0.

Algorithm 1. *Encode*(T)

```
Input: T a tree
Output: A number of bits being enough for encoding T
begin
    if T is a leaf then
        return 0
    else
        for x_i ∈ Children(T) do
            s_i = Encode(x_i)
        return W(S =< s_1, ...s_n >)
end
```

The crucial point of this algorithm lives in the function noted by $\mathcal{W}()$. For an orderly sequence of integers $S =< s_1, ..., s_n >$ which corresponds to the weight of the children of a given node, the $\mathcal{W}()$ function returns a number of bits necessary to encode it. In the following we give several manners to implement the $\mathcal{W}()$ function.

The first one corresponds to the Dichotomic Encoding [32]. This strategy consists in determining the weight of a sequence S according to its number of elements. If the sequence S contains only one element, S weighs one more than this unique element. In the case of S contains two elements its weight is two more than the largest element of S. On the other hand if S contains at most three elements the Dichotomic Encoding selects the two smallest elements (*i.e.,* s_1 and s_2), and replaces them by an element weighing $s_2 + 2$, and iterates this process.

More formally the behaviour of the Dichotomic Encoding is given by the following recursive function

$$\mathcal{D}(S) = \begin{cases} 0 & |S| = 0 \\ s_1 + 1 & |S| = 1 \\ s_2 + 2 & |S| = 2 \\ \mathcal{D}(< s_3, ..., s_2 + 2, ..., s_n >) & |S| > 2 \end{cases}$$

The idea of Filman's Polychotomic Encoding rests in the following observation. A sequence $S =< s_1, s_2, ..., s_n >$, where the distance between s_2 and s_n is strictly lower than 2, is called flat sequence. Filman has shown that for a flat sequence we have $s_n + sp(n) \leq \mathcal{D}(S)$. Consequently, in case of flat sequence it's always better to use $s_n + sp(n)$ bits rather than the encoding given by the Dichotomic Encoding. In all other case Polychotomic Encoding works as the Dichotomic Encoding. Its functioning is formalized by the following.

$$\mathcal{P}(S) = \begin{cases} 0 & |S| = 0 \\ s_1 + 1 & |S| = 1 \\ s_2 + 2 & |S| = 2 \\ \mathcal{P}(< s_3, ..., s_2 + 2, ..., s_n >) & |S| > 2 \text{ and } s_n - s_2 \geq 2 \\ s_n + sp(n) & |S| > 2 \text{ and } s_n - s_2 < 2 \end{cases}$$

In the following we give a new heuristic based on a generalization of the Filman's observation.

3 Heuristic

3.1 Description

In this section we present a new improvement for sequence encoding. On the principle of Filman's strategy which uses $s_n + sp(n)$ bits to encode a flat sequence. We can wonder if we cannot use a similar method when the sequence presents landings. Landings can be seen as a flat subsequence. As example the following sequence owns two landings, one between s_1 and s_5, and another one between s_6 and s_9. Indeed, one can check that $s_5 - s_2 \leq 1$ and $s_9 - s_7 \leq 1$.

$$< 1, 2, 2, 3, 3, 8, 8, 9, 9, 14, 19 >$$

Our new heuristic called *Generalized Polychotomic Encoding* detects such landings (*i.e.,* $s_k - s_2 < 2$), and decides if it's interesting to merging it. It's interesting to merge a landing of size k when the distance between elements s_k and s_{k+1} are big enough (*i.e.,* $s_k + sp(k) \leq s_{k+1}$). In all other cases Generalized Polychotomic Encoding works as the Polychotomic Encoding. Generalized Polychotomic Encoding can be formalized as follow

$$\mathcal{G}(S) = \begin{cases} 0 & |S| = 0 \\ s_1 + 1 & |S| = 1 \\ s_2 + 2 & |S| = 2 \\ s_n + sp(n) & |S| \geq 2 \text{ and } s_n - s_2 < 2 \\ \mathcal{G}(< s_k + sp(k), s_{k+1}, ..., s_n >) & |S| \geq 2 \text{ and } \exists k \leq n-1 \\ & \text{with } \ s_k - s_2 < 2, s_k + sp(k) \leq s_{k+1} \\ \mathcal{G}(< s_3, ..., s_2 + 2, ..., s_n >) & \text{in other cases} \end{cases}$$

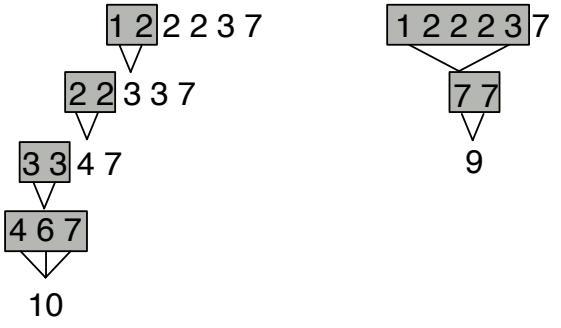


Fig. 2. On left, Polychotomic Encoding computation induces three Dichotomic steps and ends with the Filman's step. $\mathcal{P}(< 1, 2, 2, 2, 3, 7 >) = 10$. On right our technique detects the flat subsequence $< 1, 2, 2, 2, 3 >$ and merge it to obtain a 7. We then obtain $\mathcal{G}(< 1, 2, 2, 2, 3, 7 >) = 9$.

The following example shows the interest of Generalized Polychotomic Encoding on the sequence $< 1, 2, 2, 2, 3, 7 >$.

If the $sp(k)$ values have been precomputed and can be accessed in constant time, the computation of $\mathcal{G}(S)$ from S has a linear time complexity.

3.2 Theoretical Results

Let S be a sequence, then Generalized Polychotomic Encoding is never worse than Polychotomic Encoding. In other words we have

Proposition 4. *For any sequence S, $\mathcal{G}(S) \leq \mathcal{P}(S)$.*

See Appendix for the proof.

Let S a well-chosen sequence, the difference between the respective sizes of the Polychotomic Encoding of S and the Generalized Polychotomic Encoding of S can be as large as wished. In other words we have.

Proposition 5. *For any positive integer n, there exists a sequence S such that $\mathcal{P}(S) - \mathcal{G}(S) \geq n$.*

Proof. Let us consider the following lemma (the proof is given in Appendix):

Lemma 1. *Let $S =< 0, 0, ..., 0, sp(k) >$ be a sequence of size $k+1$ with $k = 2^p$. Then we have*

- $\mathcal{G}(< 0, 0, ..., 0, sp(k) >) = \mathcal{G}(< sp(k), sp(k) >) = sp(k) + 2$;
- $\mathcal{P}(< 0, 0, ..., 0, sp(k) >) \geq sp(k) + sp(2^{p-(sp(k)/2)})$;

Let $\Delta = \mathcal{P}(S) - \mathcal{G}(S)$. From lemma 1 we obtain $\Delta = sp(2^{p-(sp(k)/2)}) - 2$. Let us now show that the difference between $\mathcal{P}(S)$ and $\mathcal{G}(S)$ growths with k. From Proposition 3 in one hand we have :

$$\begin{aligned} \Delta &\geq log_2(2^{p-(sp(k)/2)}) - 2 \\ &\geq p - sp(k)/2 - 2 \end{aligned} \tag{1}$$

on the other hand we obtain :

$$\begin{aligned} sp(k) = sp(2^p) &\leq log_2 2^p + (log_2 log_2 2^p)/2 + 2 \\ &\leq p + (log_2 p)/2 + 2 \end{aligned} \tag{2}$$

Form equations 1 and 2 we conclude

$$\begin{aligned} \Delta &\geq p - [(p/2) + (log_2 p)/4 + 1) - 2 \\ &\geq p/2 - (log_2 p)/4 - 3 \end{aligned} \tag{3}$$

As example Δ becomes positive with $p = 8$ and Δ is equal to 3 with $p = 13$.

4 Experimental Results

We're going to evaluate experimentally our algorithm on "natural" hierarchies, and on random trees. This experiment aim is to compare Dichotomic Encoding, Polychotomic Encoding, and Generalized Polychotomic Encoding.

The natural hierarchies mainly come from programming languages (Classes and Packages hierarchies) and Artificial Intelligence purpose. As classes hierarchies the following examples are shown: VisualWorks2, NeXTStep, Digitalk3, ET++. Those examples are the benchmarks of [30,32,27]. We use the package hierarchies of Java 1.3 and JavaSE6. As example of an AI's hierarchy our heuristic has been applied to a biological taxonomy named Mammals. Table 1 presents the number of bits required to encode that trees.

The random trees are produced by using the following protocol: we fix the maximal height $hMax$ and the maximum branching factor $bMax$ as two parameters. From the root, we randomly choose its number of children (uniformly between 1 and bMax). And for each of its children we choose uniformly a boolean which determined if it is a leaf or not. This process iterates until it reaches the maximal height. The experimental study carries on 50000 trees to avoid special instances. Table 2 presents the average number of nodes and the averages of the encoding for every

Table 1. Generalized Polychotomic Encodings for the benchmarks of [30,32,27]

	$\mathcal{D}$	$\mathcal{P}$	$\mathcal{G}$
Class			
VisualWorks2	32	20	19
Digitalk3	27	26	26
NeXTStep	18	17	17
ET++	20	20	19
Packages			
Java 1.3	27	23	23
JavaSE 6	29	26	26
AI			
Mammals	30	26	25

Table 2. Random trees are defined by a maximal height ("hMax"), a branching factor ("bMax") and a number of nodes ("Nodes"). This table summerizes the value of the different functions $\mathcal{D}$, $\mathcal{P}$ and $\mathcal{G}$ for each kind of tree.

hMax	bMax	Nodes	$\mathcal{D}$	$\mathcal{P}$	$\mathcal{G}$	Distance
5	20	53,5	15,9	14,2	13,4	6%
	40	103,2	18	16	14,9	7,4%
	200	503,5	22,7	20	18,1	10,5%
	500	1252,8	25,2	22,3	19,6	13,8%
	1000	2498,7	27,2	24	20,8	15,4%
10	20	106,1	25,6	24	23,8	0,8%
	40	205,9	27,9	25,9	24,8	4,4%
	200	1004,8	32,7	30	28	7,1%
	500	2509,9	35,2	32,3	29,7	8,8%
	1000	5011,8	37,2	34	30,9	10%
20	20	211,5	45,1	43,1	42,7	1,9%
	40	411,1	47,6	45,7	44,6	2,5%
	200	2008,8	52,6	50	48	4,2%
	500	5005,8	55,1	52,3	49,6	5,4%
	1000	9983,1	57,2	54	50,8	6,3%

couple of parameters. As well as the average distance between the Polychotomic Encoding and the Generalized Polychotomic Encoding.

Our different experimental studies enlighten us on several points. First, we can remark that Generalized polychotomic Encoding rarely improves the polycotomic Encoding on natural hierarchies. From this observation we can deduced that landings appear rarely in nature. Second, experiments on random trees show that improvement of Generalized polycotomic Encoding compared to filman's encoding is proportional to the ratio between maximal branching factor and the maximal height. In other words, the more the considered tree is crushed, the more our strategy improves that of Filman. Since, crushed trees are convenient in appearances of landings.

5 Conclusion

Although the complexity of computing the exact 2-dimension of trees remains open, we have provided a new heuristic which produces the smallest bit-vector encodings of trees at present, and thus a better upper bound on the 2-dimension of trees.

References

1. Habib, M., Nourine, L., Raynaud, O., Thierry, E.: Computational aspects of the 2-dimension of partially ordered sets. Theor. Comput. Sci. 312(2-3), 401–431 (2004)
2. Agrawal, R., Borgida, A., Jagadish, J.V.: Efficient management of transitive relationships in large data and knowledge bases. In: ACM SIGMOD International Conference on Management of Data, pp. 115–146 (1989)

3. Schubert, L.K., Papalaskaris, M.A., Taugher, J.: Determining type, part, color and time relationships. Computer 16, 53–60 (1983)
4. Cohen, N.H.: Type-extension type tests can be performed in constant time. ACM Transactions on Programming Languages and Systems 13(4), 626–629 (1991)
5. Fall, A.: The foundations of taxonomic encodings. Computational Intelligence 14, 598–642 (1998)
6. Vitek, J., Horspool, R., Krall, A.: Efficient type inclusion tests. In: OOPSLA 1997, pp. 142–157 (1997)
7. Bouchet, A.: Etude combinatoire des ordonnés finis, Applications. PhD thesis, Université scientifique et médicale de Grenoble (1971)
8. Mattern, F.: Virtual time and global states in distributed systems. In: Parallel and Distributed Algorithms, pp. 215–226. Elsevier, Amsterdam (1989)
9. Mehlhorn, K.: Data Structures and Algorithms 2: Graph Algorithms and NP-completness. EATCS Monographs on Theoretical Computer Science. Springer, Heidelberg (1984)
10. Simon, K.: An improved algorithm for transitive closure on acyclic digraphs. Theoretical Computer Science 58, 325–346 (1988)
11. Gil, J., Zibin, Y.: Efficient subtyping tests with pq-encoding. ACM Trans. Program. Lang. Syst. 27(5), 819–856 (2005)
12. Zibin, Y., Gil, Y.: Efficient subtyping tests with pq-encoding. In: Proceedings of OOPSLA 2001 (2001)
13. Alon, N., Scheinerman, E.R.: Degrees of freedom versus dimension for containment orders. Order (5), 11–16 (1988)
14. Fishburn, P., Trotter, T.: Geometric containment orders: a survey. Order (15), 167–182 (1999)
15. Capelle, C.: Representation of an order as union of interval orders. In: Bouchitté, V., Morvan, M. (eds.) ORDAL 1994. LNCS, vol. 831, pp. 143–161. Springer, Heidelberg (1994)
16. West, D.B.: Parameters of partial orders and graphs: packing, covering and representation. In: Graphs and Orders, NATO, pp. 267–350. D. Reidel publishing company (1985)
17. Dahl, V., Fall, A.: Logical encoding of conceptual graph type lattices. In: First International Conference on Conceptual Structures, Canada, pp. 216–224 (1993)
18. Gambosi, G., Nesestril, J., Talamo, M.: Posets, boolean representation and quick path searching. In: Proceedings of ICALP 1987, pp. 404–424 (1987)
19. Gambosi, G., Nesestril, J., Talamo, M.: On locally presented posets. Theoretical Comp. Sci. 70(2), 251–260 (1990)
20. Gambosi, G., Nesestril, J., Talamo, M.: Efficient representation of taxonomies. In: Proceedings of TAPSOFT 1987, pp. 232–240 (1987)
21. At-Kaci, H., Boyer, R., Lincoln, P., Nasr, R.: Efficient implementation of lattice operations. ACM Transactions on Programming Languages and Systems 11(1), 115–146 (1989)
22. Talamo, M., Vocca, P.: An efficient data structure for lattice operations. SIAM J. Comput. 28(5), 1783–1805 (1999)
23. Habib, M., Huchard, M., Nourine, L.: Embedding partially ordered sets into chain-products. In: Proceedings of KRUSE 1995, pp. 147–161 (1995)
24. Trotter, W.T.: Combinatorics and Partially Ordered Sets: Dimension Theory. John Hopkins University Press, Baltimore (1991)
25. Ellis, G.: Efficient retrieval from hierarchies of objects using lattice operations. In: Conceptual graphs for knowledge representation (International Conference on Conceptual Structures). LNCS (LNAI), vol. 699. Springer, Heidelberg (1993)

26. Novak, V.: On the pseudo-dimension of ordered sets. Czechoslovak Math. Journal 13, 587–598 (1963)
27. Filman, R.E.: Polychotomic encoding: A better quasi-optimal bit-vector encoding of tree hierarchies. In: Magnusson, B. (ed.) ECOOP 2002. LNCS, vol. 2374, pp. 545–561. Springer, Heidelberg (2002)
28. Engel: Sperner Theory. Cambridge University Press, Cambridge (1997)
29. Caseau, Y.: Efficient handling of multiple inheritance hierarchies. In: Proceedings of OOPSLA 1993, pp. 271–287 (1993)
30. Krall, A., Vitek, J., Horspool, R.: Near optimal hierarchical encoding of types. In: Proceedings of ECOOP 1997, pp. 128–145 (1997)
31. Caseau, Y., Habib, M., Nourine, L., Raynaud, O.: Encoding of multiple inheritance hierarchies and partial orders. Computational Intelligence 15, 50–62 (1999)
32. Raynaud, O., Thierry, E.: A quasi optimal bit-vector encoding of tree hierarchies. application to efficient type inclusion tests. In: Knudsen, J.L. (ed.) ECOOP 2001. LNCS, vol. 2072, pp. 165–180. Springer, Heidelberg (2001)

Mathematical Programming Formulations for the Bottleneck Hyperplane Clustering Problem

Kanika Dhyani[1,2] and Leo Liberti[1]

[1] LIX, École Polytechnique, 91128 Palaiseau, France
{dhyani,liberti}@lix.polytechnique.fr
[2] DEI, Politecnico Di Milano, P.zza L. Da Vinci 32, 20133 Milano, Italy
dhyani@elet.polimi.it

Abstract. We discuss a mixed-integer nonlinear programming formulation for the problem of covering a set of points with a given number of slabs of minimum width, known as the bottleneck variant of the hyperplane clustering problem. We derive several linear approximations, which we solve using a standard mixed-integer linear programming solver. A computational comparison of the performance of the different linearizations is provided.

Keywords: MINLP, k-line center problem, reformulation, linearization.

1 Introduction

We investigate some mathematical programming formulations for the following optimization problem.

BOTTLENECK HYPERPLANE CLUSTERING PROBLEM (bHCP). Given integers $n, m, d > 0$ and a set $N = \{\boldsymbol{p}_i \in \mathbb{R}^d \mid i \leq n\}$, find a set $M = \{(\boldsymbol{w}_j, w_j^0) \in \mathbb{R}^d \times \mathbb{R} \mid j \leq m\}$ and an assignment $x : N \times M \to \{0, 1\}$ of points to hyperplanes such that $\max\limits_{\substack{i \leq n, j \leq m \\ x_{ij}=1}} \frac{|\boldsymbol{w}_j \cdot \boldsymbol{p}_i - w_j^0|}{||\boldsymbol{w}_j||_2}$ is minimum.

In other words, we want to partition N into m clusters whose points are projected onto a $(d-1)$-dimensional subspace in such a way that the maximum Euclidean distance between a point and its projection is minimized. Our problem is a special case of a projective clustering problem in which all the subspaces are of the same dimension. It is also known as the HYPERPLANE COVER PROBLEM [8], the m-HYPERPLANE CENTER PROBLEM [14] and the SLAB WIDTH PROBLEM [6] in literature.

If we fix $\max\limits_{\substack{i \leq n, j \leq m \\ x_{ij}=1}} \frac{|\boldsymbol{w}_j \cdot \boldsymbol{p}_i - w_j^0|}{||\boldsymbol{w}_j||_2}$ to some maximum tolerance, geometrically our problem is that of finding slabs of minimum width that cover all the points — thus the name bottleneck. In the case when the slabs are of zero width, the problem at hand is known as the k-line center problem in which lines are used instead of slabs. The k-line center problem has been studied extensively in

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 87–96, 2008.

literature: [14] reports a table with summarized complexities of the developed algorithms. Most of the past work studied the problem from a computational geometry point of view; most results are theoretical in nature. To the best of our knowledge, mathematical programming based solution approaches for the bHCP have not been extensively studied yet.

1.1 Previous Work

Clustering techniques are widely studied in areas ranging from data mining to information retrieval and pattern recognition to name a few. They also arise in the context of shape fitting in geometric covering problems where given a set of shapes the goal is to choose the one that covers the data points w.r.t. some objective. Deciding whether a set of n points in $\mathbb{R}^2$ can be covered with m lines was shown to be **NP**-complete in [13]; trying to approximate the width of the minimum slab that covers the point is also **NP**-complete.

A sublinear time randomized algorithm in which all but $(1-\gamma)n$ of the points are covered is presented in [14]. They prove that the m-q-dimensional hyperplane center problem (where q is the dimension of the subspace) can be solved in $\tilde{\mathcal{O}}(d\frac{mq}{\gamma}^{q+1})$. The described algorithm finds a collection of $\mathcal{O}(m \log \frac{mdq}{\gamma})$ slabs of width at most 2^q times the optimum.

For points in $\mathbb{R}^2$ and $\mathbb{R}^3$, [1] present randomized algorithms which compute $\mathcal{O}(m \log m)$ strips of bounded width that cover the data points. Their algorithms have run times of $\mathcal{O}(nm^2 \log^4 n)$ if $m^2 \log m \leq n$ and $\mathcal{O}(n^{2/3}m^{8/3}\log^4 n)$ for larger m when $d = 2$ and $\mathcal{O}(n^{3/2}m^{9/4}\text{polylog}(n))$ for $d = 3$.

A coreset framework based approach was proposed in [6]. It was shown that no coreset exists for the problem of covering a set of points with 2 slabs in $\mathbb{R}^3$; however, a $(1+\gamma)$ approximation algorithm for the same problem was also presented.

In [8], some fixed-parameter tractable algorithms which are based on general techniques of parameterized complexity are presented. The main argument of this work rests on the fact that certain parameters (such as the dimension) can be used to limit the complexity of these problems.

Many variants of the bHCP have been proposed in the literature. [4] adapted the k-means algorithm to the case of hyperplanes and treated the problem of minimizing the sum of the Euclidean distances of points to the assigned hyperplanes (for fixed m). Another variant minimizes the total number of hyperslabs of a fixed width used to cover a set of points [2].

The rest of this paper is organized as follows. The mathematical programming formulation of the bHCP is given in Sect. 2. Some exact reformulations are given in Sect. 3. Three model-based approximating linearizations are proposed in Sect. 4. Our computational results are discussed in Sect. 5. Sect. 6 concludes the paper.

2 Problem Formulation

Given a set of n points $\boldsymbol{p}_i \in \mathbb{R}^d$ $(i \leq n)$ we seek a set of m hyperplanes $(\boldsymbol{w}_j, w_j^0) \in \mathbb{R}^{d+1}$ $(j \leq m)$ and an assignment $\boldsymbol{x} \in \{0,1\}^{nm}$ (such that $x_{ij} = 1$ iff

$\boldsymbol{p}_i$ is assigned to the hyperplane $(\boldsymbol{w}_j, w_j^0)$ for all $i \leq n, j \leq m$) that minimizes the maximum of the Euclidean distances between the hyperplanes and their assigned points. The following Mixed-Integer Nonlinear Programming (MINLP) formulation correctly describes the problem:

$$\left.\begin{array}{rl} \min \max\limits_{\substack{j\leq m\\ i\leq n}} \frac{|\boldsymbol{w}_j\boldsymbol{p}_i - w_j^0|}{||\boldsymbol{w}_j||_2} x_{ij} & \\ \text{s.t. } \forall i \leq n & \sum\limits_{j\leq m} x_{ij} = 1 \\ & \boldsymbol{w} \in \mathbb{R}^{md} \\ & \boldsymbol{x} \in \{0,1\}^{nm}. \end{array}\right\} \tag{1}$$

Computationally, the direct solution of (1) is problematic because (1) is non-differentiable and has a singularity at $\boldsymbol{w} = 0$.

3 Reformulations

Reformulations are symbolic transformations that are applied to the problem formulation and yield modified formulations with different mathematical properties [10]. Within this paper, we shall make use of two types of reformulations: opt-reformulations and approximations. *Opt-reformulations* guarantee that there is an optimum of the reformulated problem corresponding to each optimum of the original problem [11]. The precise definition of an approximation is given in [11]; for our purposes, it suffices to know that an approximating reformulation yields a sequence of reformulated problems dependent on a numerical parameter, which "tends" to the original problem when the parameter tends to infinity. An approximation is simply a problem in the sequence.

We first provide opt-reformulations that yield a differentiable MINLP. We remark that if we require all vectors $\boldsymbol{w}$ to be normalized to 1, there is no need for dividing the objective function terms through by $||\boldsymbol{w}||_2$: the objective thus becomes

$$\min \max_{\substack{j\leq m\\ i\leq n}} |\boldsymbol{w}_j\boldsymbol{p}_i - w_j^0| x_{ij}$$

subject to added constraints

$$\forall j \leq m \quad ||\boldsymbol{w}_j||_2^2 = 1. \tag{2}$$

We reformulate the maximum operator by introducing an added nonnegative continuous variable $\varepsilon \geq 0$: the objective becomes

$$\min \varepsilon$$

subject to added constraints

$$\forall i \leq n, j \leq m \quad \varepsilon \geq |\boldsymbol{w}_j\boldsymbol{p}_i - w_j^0| x_{ij}.$$

Secondly, we reformulate the absolute values by introducing added nonnegative continuous variables $t^+_{ij}, t^-_{ij} \geq 0$, which yield reformulated constraints

$$\forall i \leq n, j \leq m \quad \varepsilon \geq (t^+_{ij} + t^-_{ij})x_{ij}, \tag{3}$$

subject to added constraints

$$\forall i \leq n, j \leq m \quad \boldsymbol{w}_j \boldsymbol{p}_i - w^0_j = t^+_{ij} - t^-_{ij}.$$

This reformulation is exact as long as a complementarity constraint $\sum_{i,j} t^+_{ij} t^-_{ij} = 0$ is enforced; in this particular case, however, it is not necessary because of the minimization direction of the objective function. Lastly, since the products $t^+_{ij}x_{ij}$ and $t^-_{ij}x_{ij}$ involve a binary variable, they can be linearized exactly by replacing them with added nonnegative continuous variables y^+_{ij}, y^-_{ij} whilst adding the following (linear) constraints:

$$\begin{aligned}
\forall i \leq n, j \leq m \quad & y^+_{ij} \leq \min(Mx_{ij}, t^+_{ij}) \\
\forall i \leq n, j \leq m \quad & y^+_{ij} \geq t^+_{ij} - M(1 - x_{ij}) \\
\forall i \leq n, j \leq m \quad & y^-_{ij} \leq \min(Mx_{ij}, t^-_{ij}) \\
\forall i \leq n, j \leq m \quad & y^-_{ij} \geq t^-_{ij} - M(1 - x_{ij}),
\end{aligned}$$

where M is a large enough constant. We also remark that the intuitive meaning of (3) is that ε should be bounded below by $t^+_{ij} + t^-_{ij}$ if and only if $x_{ij} = 1$. This can be written formally as:

$$\forall i \leq n, j \leq m \quad y_{ij} \geq t^+_{ij} + t^-_{ij},$$

where y_{ij} is an added nonnegative continuous variable constrained to take value ε if and only if $x_{ij} = 1$ (otherwise, it is free):

$$\begin{aligned}
\forall i \leq n, j \leq m \quad & y_{ij} \leq \varepsilon + M(1 - x_{ij}) \\
\forall i \leq n, j \leq m \quad & y_{ij} \geq \varepsilon - M(1 - x_{ij}).
\end{aligned}$$

The latter approach provides an alternative linearization of the products.

The reformulations above therefore provide two different twice-differentiable MINLP formulations for the bHCP:

$$\left.\begin{aligned}
\min \quad & \varepsilon \\
\text{s.t.} \forall i \leq n, j \leq m \quad & \varepsilon \geq y^+_{ij} + y^-_{ij} \\
\forall j \leq m \quad & ||\boldsymbol{w}_j||^2_2 = 1 \\
\forall i \leq n, j \leq m \quad & \boldsymbol{w}_j \boldsymbol{p}_i - w^0_j = t^+_{ij} - t^-_{ij} \\
\forall i \leq n, j \leq m \quad & y^+_{ij} \leq \min(Mx_{ij}, t^+_{ij}) \\
\forall i \leq n, j \leq m \quad & y^+_{ij} \geq t^+_{ij} - M(1 - x_{ij}) \\
\forall i \leq n, j \leq m \quad & y^-_{ij} \leq \min(Mx_{ij}, t^-_{ij}) \\
\forall i \leq n, j \leq m \quad & y^-_{ij} \geq t^-_{ij} - M(1 - x_{ij}) \\
\forall i \leq n \quad & \sum_{j \leq m} x_{ij} = 1 \\
& \boldsymbol{w} \in \mathbb{R}^{md} \\
& \boldsymbol{x} \in \{0, 1\}^{nm} \\
& \boldsymbol{y}^+, \boldsymbol{y}^-, \boldsymbol{t}^+, \boldsymbol{t}^- \in [0, M]^{nm}
\end{aligned}\right\} \tag{4}$$

$$\left.\begin{array}{rrl} \min & \varepsilon & \\ \text{s.t.} \forall i \leq n, j \leq m & y_{ij} \geq t^+_{ij} + t^-_{ij} & \\ \forall j \leq m & ||\boldsymbol{w}_j||_2^2 = 1 & \\ \forall i \leq n, j \leq m & \boldsymbol{w}_j \boldsymbol{p}_i - w^0_j = t^+_{ij} - t^-_{ij} & \\ \forall i \leq n, j \leq m & y_{ij} \leq \varepsilon + M(1 - x_{ij}) & \\ \forall i \leq n, j \leq m & y_{ij} \geq \varepsilon - M(1 - x_{ij}) & \\ \forall i \leq n & \sum_{j \leq m} x_{ij} = 1 & \\ & \boldsymbol{w} \in \mathbb{R}^{md} & \\ & \boldsymbol{x} \in \{0,1\}^{nm} & \\ & \boldsymbol{y}, \boldsymbol{t}^+, \boldsymbol{t}^- \in [0, M]^{nm} & \end{array}\right\} \quad (5)$$

Proposition 1. *If $(\boldsymbol{w}^*, \boldsymbol{x}^*)$ is a global optimum of (4) (resp. (5)) then it is also a global optimum of (1).*

Proof. This follows by Defn. 2.3.10 and Lemma 2.3.11 in [10], because all the reformulations involved are opt-reformulations.

Both (4) and (5) are extremely difficult problems to solve, due to the high number of binary variables and the nonconvexity of (2). Exact solutions of such MINLPs can be obtained via the spatial Branch-and-Bound (sBB) algorithm [9] only for very small instances ($\leq$ 10 points, $\leq$ 3 hyperplanes, $\leq$ 2 dimensions). MINLP heuristics such as VNS [12] fare slightly better but are far from being able to tackle realistically-sized instances.

4 Approximations

In this section we propose three different Mixed-Integer Linear Programming (MILP) approximations for the problematic nonconvex constraints (2) in terms of the ℓ_1 and ℓ_∞ norm, which can both be linearized exactly. We first remark the following inclusion relationships:

$$U_1 = \{\boldsymbol{w} \mid ||\boldsymbol{w}||_1 \leq 1\} \subseteq \{\boldsymbol{w} \mid ||\boldsymbol{w}||_2 \leq 1\} = U_2$$
$$U_2 = \{\boldsymbol{w} \mid ||\boldsymbol{w}||_2 \leq 1\} \subseteq \{\boldsymbol{w} \mid ||\boldsymbol{w}||_\infty \leq 1\} = U_\infty.$$

We shall exploit these inclusions to derive exactly linearizable approximations for U_2.

In the rest of this section, we shall discuss the exact linearization of both ℓ_1 and ℓ_∞ unit constraints. We shall then propose three different approximations of (2): the ℓ_∞ approximation, the ℓ_1/ℓ_∞ "sandwiching" approximation, and the ℓ_1/ℓ_∞ "alternating" approximation (shown graphically in Fig. 1).

4.1 Linearization of ℓ_1 Unit Constraint

The linearization of the ℓ_1 unit constraint:

$$\forall j \leq m \quad ||\boldsymbol{w}_j||_1 = 1, \quad (6)$$

which can also be written as $\sum_{k \leq d} |w_{jk}| = 1$ for $j \leq m$, proceeds by repeated application of the AbsDiff opt-reformulation [10] to each absolute value term $|w_{jk}|$: let w^+_{jk}, w^-_{jk} be added nonnegative continuous variables, replace (6) with:

$$\forall j \leq m \quad \sum_{k \leq d} (w^+_{jk} + w^-_{jk}) = 1,$$

add the constraints:

$$\forall j \leq m, k \leq d \quad w_{jk} = w^+_{jk} - w^-_{jk},$$

and add the following exact reformulation of the linear complementarity conditions $w^+_{jk} w^-_{jk} = 0$ (for $j \leq m, k \leq d$):

$$\forall j \leq m, k \leq d \quad w^+_{jk} \leq M \mu_{jk}$$
$$\forall j \leq m, k \leq d \quad w^-_{jk} \leq M(1 - \mu_{jk}),$$

where for $j \leq m, k \leq d$ μ_{jk} are added binary variables that are 1 if w^+_{jk} has nonzero value.

4.2 Linearization of ℓ_∞ Unit Constraint

The linearization of the ℓ_∞ unit constraint:

$$\forall j \leq m \quad ||\boldsymbol{w}_j||_\infty = 1, \tag{7}$$

which can also be written as $\max_{k \leq d} |w_{jk}| = 1$ for $j \leq m$, is a reformulation of the *narrowing* type [11] denoted by InfNorm, and was proposed in [5] (informally, a reformulation is a narrowing if it preserves at least one global optimum of the original problem). In full generality it works as follows. Consider a mathematical programming formulation P with a d-vector of variables $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ bounded in $[-\alpha, \alpha]$ (for some $\alpha > 0$) with the property that if x^* is a feasible solution of P then $-x^*$ is also a feasible solution of P with the same objective function cost; and a constraint $||x||_\infty = \alpha$. The InfNorm reformulation is as follows:

- for all $k \leq d$, add a binary decision variable u_k to P;
- delete the constraint $||x||_\infty = \alpha$;
- add the following constraints:

$$\forall k \leq d \quad x_k \geq \alpha(1 - 2(1 - u_k)) \tag{8}$$
$$\sum_{k \leq d} u_k = 1. \tag{9}$$

This reformulation being a narrowing, it guarantees that at least one optimum of the original problem is mapped into an optimum of the reformulation.

Proposition 2. *There exist an optimum of P which is also an optimum of* InfNorm(P).

Proof. Constraint (9) ensures that there is at least an index $k \leq d$ such that, by (8) and the fact that the upper bound for x_k is α, the value of x_k is exactly α: this forces $||x||_\infty$ to be precisely α. Suppose now there is a feasible values of x with $||x||_\infty = \alpha$ such that $x_k \neq \alpha$ for all k. Since $||x||_\infty = \alpha$, this implies there is at least an index $k \leq d$ for which $x_k = -\alpha$. By the symmetry assumption, $-x$ is feasible and has the same objective function value as x.

4.3 Pure ℓ_∞ Approximation

This approximation is based on simply replacing (2) by (7) and applying the InfNorm narrowing. Geometrically, we replace a hyperspherical feasible region with its hypercubical overapproximation, as shown graphically in Fig. 1 (A).

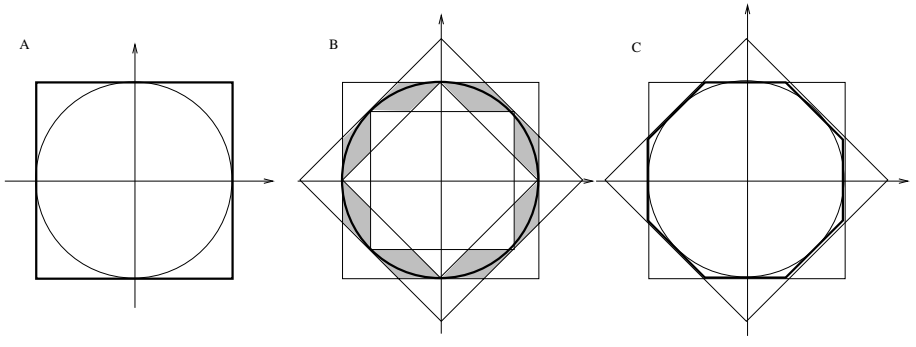


Fig. 1. Three types of approximation: pure (A), sandwich (B), and alternating (C)

An equivalent pure ℓ_1 approximation (where (2) was replaced by (6) and subsequently linearized) was tested but found to yield markedly inferior performances, and thence discarded.

4.4 Sandwiching ℓ_1/ℓ_∞ Approximation

This approximation, depicted graphically in Fig. 1 (B), consists of replacing (2) by the following constraints:

$$\forall j \leq m \quad 1 \leq ||\boldsymbol{w}_j||_1 \leq \sqrt{d}$$
$$\forall j \leq m \quad \frac{1}{\sqrt{d}} \leq ||\boldsymbol{w}_j||_\infty \leq 1.$$

The above constraints can be linearized exactly by applying the reformulations of Sections 4.1 and 4.2.

4.5 Alternating ℓ_1/ℓ_∞ Approximation

This approximation, depicted graphically in Fig. 1 (C), consists of replacing (2) by the following disjunction:

$$\forall j \leq m \quad (||\boldsymbol{w}_j||_\infty = 1 \quad \vee \quad ||\boldsymbol{w}_j||_1 = \sqrt{d}).$$

This is modelled by introducing added binary variables $\mu_j \in \{0,1\}$ for $j \leq m$ which have value 1 if the constraint $||\boldsymbol{w}_j||_\infty = 1$ is active and 0 otherwise, and serve the purpose of alternating between ℓ_1 and ℓ_∞ unit norm constraints.

5 Computational Experiments

We considered a set of 8 instances whose statistics are given in Table 1. The cancer instance is taken from the Wisconsin Prognostic Breast Cancer (WPBC) Database [3]. The other instances are generated based on hyperplane geometry in [5]. All instances were solved using 6 different MILP formulations: (4) and (5) with the "pure ℓ_∞" (Sect. 4.3), "sandwich" (Sect. 4.4) and "alternating" (Sect. 4.5) approximations. All results were obtained on one core of a quad-CPU Intel Xeon 2.4GHz with 8GB RAM using ILOG CPLEX 11.0 [7].

Table 1. Instance statistics

Instance	n	m	d
cancer	194	3	2
esr_150_8_10	150	8	10
esr_210_8_10	210	8	10
sr_2000_7_4	2000	7	4
sr_250_10_10	250	10	10
sr_500_4_4	500	4	4
sr_750_10_10	750	10	10
hsynt_1500_8_3	1500	8	2
synt_35_2_3	35	3	2
synt_70_2_3	70	3	2

The results are given in Table 2. Each group of 5 columns describes the results obtained by solving all instances in the test set with a particular formulation. Within each group, we recorded: the value of objective function of the original problem (1) $\varepsilon = \max\limits_{\substack{j\leq m\\ i\leq n}} \frac{|\boldsymbol{w}_j \boldsymbol{p}_i - w_j^0|}{||\boldsymbol{w}||_2} x_{ij}$ computed using the solution given by the approximation; the CPU time (limited to 1800 seconds of user time); the Branch-and-Bound (BB) node index F where the recorded optimum was found; the total number N of BB nodes explored within the allowed time; and the final optimality gap reported by CPLEX (opt=0%). For each approximation type, boldface is used to emphasize the original problem ((4) or (5)) yielding better results for a particular measure (for F, N the comparative measure is F/N). Underlined figures emphasize draws. Values marked by * indicate a winning method for a particular instance: this is chosen by looking at (in order of priority): ε, CPU, gap, F/N.

Table 2. Computational results

Instance	(4)+Pure ℓ_∞					(5)+Pure ℓ_∞				
	ε	CPU	F	N	gap	ε	CPU	F	N	gap
cancer	**0.73***	1800.21	677574	760654	**12.33%**	0.915	1800.11	**7671**	209071	67.65%
esr_150_8_10	**0.0744***	1800.21	13879	13980	100.00%	0.0896	1800.26	**10164**	11264	100.00%
esr_210_8_10	0.1704	1800.32	**7874**	8075	100.00%	**0.1503***	1800.34	5371	5472	100.00%
sr_2000_7_4	0.4883	1802.44	**507**	607	100.00%	**0.4339***	1802.78	534	536	100.00%
sr_250_10_10	**0.1431***	1800.43	5751	5852	100.00%	0.2219	1800.48	**1889**	1990	100.00%
sr_500_4_4	0.3755	1800.43	6080	6181	100.00%	**0.3588**	1800.28	**6260**	6561	100.00%
sr_750_10_10	0.4059	1805.49	**680**	782	100.00%	**0.3535***	1803.21	789	790	100.00%
hsynt_1500_8_3	0.4497	1800.89	**511**	812	100.00%	**0.4368**	1800.46	1911	2129	100.00%
synt_35_2_3	0.0561	**6.33**	5103	7500	opt	**0.0556**	31.53	**10211**	41062	opt
synt_70_2_3	0.0749	**47.52**	8661	10238	opt	**0.0741**	163.43	**24223**	62325	opt

Instance	(4)+Sandwich					(5)+Sandwich				
	ε	CPU	F	N	gap	ε	CPU	F	N	gap
cancer	**0.8922**	1800.07	104385	119285	**52.53%**	0.9286	1800.12	**105540**	243340	53.52%
esr_150_8_10	0.1121	1800.28	18411	18612	100.00%	**0.1006**	1800.2	8841	8942	100.00%
esr_210_8_10	0.1593	1800.56	11158	11175	100.00%	**0.157**	1800.43	**5964**	6165	100.00%
sr_2000_7_4	0.5386	1803.35	504	507	100.00%	**0.4863**	1802.85	**582**	586	100.00%
sr_250_10_10	0.2008	1800.88	**7984**	8185	100.00%	**0.1996**	1800.41	4959	5060	100.00%
sr_500_4_4	0.3842	1800.56	**14089**	14589	100.00%	**0.3743**	1800.43	4969	4969	100.00%
sr_750_10_10	0.4939	1810.65	510	675	100.00%	**0.528**	1801.53	**571**	771	100.00%
hsynt_1500_8_3	0.5103	1801.88	**860**	1962	100.00%	**0.416***	1800.6	1701	2002	100.00%
synt_35_2_3	**0.0473**	**6.7**	891	3314	opt	0.0482	127.63	**28025**	117132	0.27%
synt_70_2_3	**0.0656**	**199.78**	84489	84490	opt	0.0664	648.44	**71625**	334408	opt

Instance	(4)+Alternating					(5)+Alternating				
	ε	CPU	F	N	gap	ε	CPU	F	N	gap
cancer	**0.73**	1800.11	131488	163688	51.47%	0.82	1800.11	**168988**	269689	**50.82%**
esr_150_8_10	0.1511	1800.2	**15374**	15875	100.00%	**0.1076**	1800.26	11060	11161	100.00%
esr_210_8_10	0.2184	1800.29	14284	14685	100.00%	**0.1871**	1800.29	**7374**	7575	100.00%
sr_2000_7_4	0.5013	1801.75	**90**	113	100.00%	**0.4845**	1802.21	483	485	100.00%
sr_250_10_10	**0.2249**	1800.35	**7394**	7595	100.00%	0.2667	1800.42	4669	4770	100.00%
sr_500_4_4	0.2871	1800.49	9095	9103	100.00%	**0.242***	1800.19	**7773**	8092	100.00%
sr_750_10_10	**0.4681**	1802.23	487	489	100.00%	0.4429	1802.32	**521**	625	100.00%
hsynt_1500_8_3	0.4741	1800.75	478	879	100.00%	**0.4713**	1800.65	**501**	2204	100.00%
synt_35_2_3	0.0462	155.08	86293	90612	opt	0.0462*	**85.31**	**12201**	50181	opt
synt_70_2_3	**0.0584***	206.55	**14715**	45222	opt	0.0596	259.43	50931	73956	opt

6 Conclusion

We presented several approximate MILP formulations for the bHCP. In particular, we discussed some techniques for linearizing a unit ℓ_2 norm constraint approximately. We evaluated the performance of the linearizations on an instance test set. However due to lack of space we do not present all the results. It was seen that approximations derived from (5) were better than those derived from (4). However, there was no clear winner within that group.

Acknowledgements

This research was partially supported by "Chaire Thales" at École Polytechnique. We are grateful to E. Amaldi for suggesting some relevant citations, and to S. Coniglio for proposing the reformulation of the unit l_∞ norm constraint.

References

1. Agarwal, P.K., Procopiuc, C.M.: Approximation algorithms for projective clustering. Journal of Algorithms 46, 115–139 (2003)
2. Amaldi, E., Ceselli, A., Dhyani, K.: Column generation for the min-hyperplane clustering problem. In: 7th Cologne-Twente Workshop on Graphs and Combinatorial Optimization (2008)
3. Asuncion, A., Newman, D.J.: UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
4. Bradley, P.S., Mangasarian, O.L.: k-Plane clustering. J. of Global Optimization 16(1), 23–32 (2000)
5. Coniglio, S., Italiano, F.: Hyperplane clustering and piecewise linear model fitting. Master's thesis, DEI, Politecnico di Milano (2007)
6. Har-Peled, S.: No coreset, no cry. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS 2004. LNCS, vol. 3328, pp. 324–335. Springer, Heidelberg (2004)
7. ILOG. ILOG CPLEX 11.0 User's Manual. ILOG S.A, Gentilly, France (2008)
8. Langerman, S., Morin, P.: Covering things with things. Discrete Computational Geometry 33(4), 717–729 (2005)
9. Liberti, L.: Writing global optimization software. In: Liberti, L., Maculan, N. (eds.) Global Optimization: from Theory to Implementation, pp. 211–262. Springer, Berlin (2006)
10. Liberti, L.: Reformulation techniques in mathematical programming, Thèse d'Habilitation à Diriger des Recherches (November 2007)
11. Liberti, L.: Reformulations in mathematical programming: Definitions. In: Aringhieri, R., Cordone, R., Righini, G. (eds.) Proceedings of the 7th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, Crema, pp. 66–70. Università Statale di Milano (2008)
12. Liberti, L., Mladenović, N., Nannicini, G.: A good recipe for solving MINLPs. In: Maniezzo, V. (ed.) Proceedings of Matheuristics 2008 Conference, Bertinoro (2008)
13. Megiddo, N., Tamir, A.: On the complexity of locating linear facilities in the plane. Operations Research Letters 1, 194–197 (1982)
14. Mishra, N., Motwani, R., Vassilvitskii, S.: Sublinear projective clustering with outliers. In: 15th Annual Fall Workshop on Computational Geometry and Visualization (2005)

Constraint Propagation with Tabu List for Min-Span Frequency Assignment Problem

Mohammad Dib, Hakim Mabed, and Alexandre Caminada

UTBM, SET Lab, 90010 Belfort Cedex, France
{mohammad.dib,hakim.mabed,alexandre.caminada}@utbm.fr

Abstract. In this paper we focus on decision tree management for effectiveness enhancement of constraint programming hybrid algorithms. We propose a deterministic method that dynamically manages the cut of tree-branches and the access to the decision variables. Both principles are based on the discovery and the memorizing of variable/value associations that lead to unfeasible solutions. The work is thus about a learning system which progressively extends the algorithm knowledge on the problem and increases its effectiveness. We have tested the method on GRAPH and CELAR frequency assignment benchmarks. The current results are on the level of the best-known ones on the Min-Span problems and they improve the success percentages on most of these problems.

Keywords: Constraint propagation, Nogood Tabu list, Frequency assignment.

1 Introduction

Constraint satisfaction problems (*CSP*) [TSA93] are generally represented by a set of variables with domain definition and a set of constraints on the variables. They have the common characteristic to be strongly combinatory and they induce high algorithmic complexity as most of them are NP-complete [GAR78] [PAP94]. Many algorithms have been developed to solve the CSP; they are usually classified in two categories. On one hand, the complete methods which guarantee to obtain an optimal solution to a given problem. Their disadvantages are that the CPU time increases exponentially with the size of the problem. In addition, incomplete methods that generally are an iterative improvement process during which the algorithm seeks one assignment satisfying the greatest number of constraints. In that case, the final objective is to satisfy all constraints. Contrary to the complete approaches, the incomplete methods cannot conclude with the unfeasibility of a problem but they are fast enough to find a good solution. The complete methods are mainly based on the concept of backtracking search [TSA93]. Search is undertaken by exploring a decisional tree structure. At each iteration, the partial solution is extended gradually by binding a variable not yet instantiated to a value of its domain. If a given variable has no more coherent value, the algorithm uninstantiates (backtrack) one of the variables previously assigned. Several techniques have been proposed to improve the performances of these algorithms:

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 97–106, 2008.

- The use of heuristic rule to determine the instantiation order of the variables either in a static way using the variables degrees or dynamically during search [BAC95] ;
- Preventing in advance the blocking assignments by the use of constraint propagation mechanisms (LookAhead). The work on Forward Checking (FC) [HAR80] and the method of Maintaining Arc-Consistency (MAC) [SAB94] are in this category;
- Other methods evaluate the blocking degree of each variable in order to operate the backtracking procedure such as Conflict-Directed Backjumping (CBJ) [PRO93, CHE01] or Dynamic Backtracking method, DBT [GIN93, JUS00].

The incomplete or approximate methods based on local search, heuristics or stochastic methods have been largely used for CSP resolution. This panel includes the greedy algorithms and simplest iterative modification methods, to most sophisticated metaheuristics (Tabu Search, Genetic Algorithms, Simulated Annealing) [HAO99, JAM03]. A great interest was also carried to the filtering algorithms [DEB97; DEB98] allowing the simplification of the CSP before or during the search. The most used methods reduce the original problem in such manner to obtain an equivalent CSP in term of solutions and to conserve *the arc-consistency* property. Filtering algorithms remove the values which obviously cannot belong to the solution and thus reduce search space. Complete methods present the advantage of result accuracy (proof of unfeasibility and ability to provide the whole solutions); however they find their limits with the increase in the size and the complexity of the considered problems. In particular, in the case of NP-complete problems, the combinatory explosion of search space induces a prohibitory calculation cost. In this case, with the loss of the exact resolution, the user can then turn to the incomplete methods which prove their effectiveness in finding a good solution in a reasonable time for larger problems.

In this context, an emergent option consists in combining the two resolution paradigms in order to profit from the respective performance of each one of them. [LAM06] presents an overview of literature on hybridization between the complete and incomplete methods. [FOC02] draws up a panorama of the Local Search (LS) applications in the context of the constraint programming. [SHA98] has proposed a Local Search method using the tree search exploration to make the extended neighborhood more competitive. Other methods use the Local Search to repair the partial solution when an inconsistency is met [JUS02]. CN-Tabu method [VAS00] [DUP05] uses the Tabu concept to temporarily prevent the instantiation of a variable with an already tested value. A lot of these approaches have shown very competitive results on several problems.

In this paper, we present a new hybrid optimization algorithm to solve CSP problem, which combines mechanisms resulting from constraint propagation and Tabu Search. Very often this kind of combination results in a more effective search able to avoid or leave the zones of search space not leading to the problem resolution. But on the other hand this type of hybridization leads to stochastic method and it may affect the properties of the method to reproduce the same quality of result from one execution to another as it is for most metaheuristic. Contrary to the majority of the other hybrid methods, we have conceived a deterministic hybrid method guaranteeing a total stability of the results on the application. The method does not use Local Search

process which most of the time adds a randomized component to the method. The search is done deterministically by constraint programming technique but Tabu memories are used to help the decision making during the search and improve the system efficiency to get better results on CSP. The Tabu memory enhances the space browsing by escaping or avoiding the zones of search space not leading to the problem resolution and then avoids to reproduce subsets of assignment leading systematically to a failure (trashing phenomenon). The paper on that work is organized as follows. We describe formally the CSP problem in section 2. Then we give the general concepts of the algorithm and its working scheme in section 3. In section 4, we report the results obtained on some CSP instances corresponding to Frequencies Assignment Problem (FAP) and we make a comparison with other methods from literature. We conclude the paper with a discussion.

2 Hybrid Method with Tabu List and Constraint Propagation

2.1 Global Procedure

We propose here a constraint propagation algorithm using a dynamic backtrack process based on the concepts of *nogood* and Tabu list. The search for the solution follows a constructive process. Search starts with a consistent partial configuration otherwise an empty assignment. At each iteration, the algorithm tries to extend the current partial assignment in such manner to maintain the consistency. The instantiation of a new variable implies adding a new constraint, called decision constraint and noted $x_i = d_i$. When a decision constraint proves to be incompatible with the union of problem constraints C and the set of decision constraints, the algorithm reacts by repairing the incompatible decisions and cancels some instantiations already done. The procedure of constraint propagation is run following each extension of the current partial configuration. This procedure consists in filtering the domains of the unassigned variables in order to detect in advance the blocking situations called *deadend*. A *deadend* is detected when the set of the possible values (maintaining the consistency of the solution) of a given variable becomes empty.

After the detection of a *deadend*, the set of the incompatible instantiations leading to the raised inconsistency is marked. These incompatible assignments form a structure called a *nogood* [SCH94]. The storage of the *nogoods* in a permanent Tabu list makes it possible to prevent the re-exploration of the blocking branches in the decision tree. When a *deadend* is reached, one of the variables taking part in the *nogood* is uninstantiated on the basis of a weight associated with each decision. The procedure of uninstantiation is described in section 3.4. The value of the uninstantiated variable is then stored in a temporary Tabu list for a Tabu period computed according to the number of times that the value was assigned to the variable. This duration is used to avoid the cycles and therefore to diversify search.

Thereby the method profits from two advantages: firstly, the reduction of search space by the means of filtering and *nogoods* procedures, and secondly the handling of consistent partial solutions. From this point of view, the method is specially dedicated to the CSP. The *ConstraintSatisfaction* procedure presented below described the general scheme of the method.

```
Procedure ConstraintSatisfaction ()
Begin
1.  iter = 0;
2.  repeat
    /*Propagation*/
3.  ConstraintPropagation();
4.  if isDeadEnd()
5.          repeat
6.                  Nogood =getNogood();
7.                  AddNogoodInPermanentTabuList(Nogood);
8.                  UnassignedVariable();
9.                  AddUnassignedVariableInTemporaryTabuList();
10.                 ConstraintPropagation();
11.         until none isDeadEnd()
12. end if
13. iter = iter+1;
    /*extend the solution*/
14. ExtendSolution();
15. until FindSolution() or iter >= iterMax
end
```

2.2 Constraint Propagation

The constraint propagation consists in reducing the domain of the variables involved in constraints. The objective is then to discard the variable values that will lead necessarily to the inconsistency of the configuration during further extensions. The variable values that inevitably cause a blocking step are called *not supported values*. Constraint propagation mechanism based on the arc-consistency consists to check the support of each variable value by considering the constraints separately. Let c a constraint connecting the variables $x_{i1},\ldots, x_{ir}$ and for which the domain of an instantiated variable is restricted to the assigned value. A value d_{ip} of one uninstantiated variable x_{ip} is said not supported by arc-consistency checking of the constraint c if:

$$d_{ip} \text{ is not supported by } c \Leftrightarrow \forall d_{i1} \ldots \forall d_{ip-1} \forall d_{ip+1} \ldots \forall d_{ik}, c \text{ is not satisfied} \quad (1)$$

$$d_{ip} \text{ is not supported} \Leftrightarrow \forall d_{i1} \ldots \forall d_{ip-1} \forall d_{ip+1} \ldots \forall d_{ik}, \exists c \in C \text{ not satisfied} \quad (2)$$

Constraint propagation procedure associates to each free variable a set of instantiated variables that have contributed to its domain restriction. The set of these variables and their values in the partial configuration will constitute the returned *nogood* if a *deadend* occurs on this variable (empty domain). For each variable of the problem the algorithm associates two domains: one current domain that represent the dynamic domain of the variable after filtering, and one original domain that corresponds to the initial domain of the variable. The idea is then to reduce the domain of the variables not instantiated taking into account: the values of assigned variables, and the incidence of the reduction of free variables domain on the domain of the other variables. The algorithm keeps trace of the origins of the reductions made on the domains. For instance, if the value d_x assigned to a variable x is in conflict with the value d_y of a free variable y then d_y will be discarded. Consequently, the variable x is both recorded as a cause of the y domain reduction and the cause of the elimination of d_y from y domain. On the other hand, if a value d_y is not supported in the filtered domain of x but is supported in the original domain by a value d_x, the value d_y is then eliminated

and the algorithm records that the reduction is due to the same reasons which leads to the elimination of d_x.

2.3 Nogood Management

In order to prevent the re-exploration of the blocking branches, the list of *nogoods* that have led to *deadend* is stored. One *nogood* is the set of the couples variable/value which has involved the domain of a variable to be empty. This unit is thus associated with a failure of the algorithm. This failure corresponds to the assignments of some variables which finally correspond to a zone of the search space that cannot contain a solution to the problem. When such a unit is identified by the algorithm it is thus important to memorize it in order not to reproduce it. In some manner this list capitalizes the search experience. The algorithm always takes care that the current partial configuration does not contain a list of assignments recorded as *nogood*. Each time a *deadend* is reached, the list of the decisions having led to the empty field is added to the list of *nogoods*. We notice that, if the decisions of a *nogood* are included in another already stored *nogood* only the minimal *nogood* is retained. All elements of that list are definitively exclude of the search. In this way the algorithm stores the sets of variable assignments of different sizes in a permanent Tabu list. This list is the fist one uses by the algorithm, a second list is also used by the uninstantiation procedure.

2.4 Uninstantiation Procedure

A weight is associated to each couple variable/value involves in the *deadends* found by the algorithm. The weight measures the importance of the couple in the previous failures. The weight is calculated in a dynamic way during the search. After each *nogood* detection, the weight of the couples variable/value corresponding to the decisions contained in the *nogood* is incremented according to the following formula:

$$Weigth\ (x,\ d_x)\ =\ Weigth\ (x,\ d_x)+1/Length\ Of\ Nogood \tag{3}$$

The length of the *nogood* corresponds to the number of variables which it contains, it is then the cardinality of the current *nogood* set. The initial value of the weight is put to null for each couple variable/value. After each *deadend* the decision with the greatest weight in the current *nogood*, is cancelled (uninstantiation of the variable). The cancelled decision is then considered Tabu for a given number of iterations in order to avoid search cycles. The Tabu duration is calculated according to the number of times that the same decision was undertaken by search. This second Tabu list is then a temporary Tabu list with full deterministic procedure that allows the algorithm to diversify its search by avoiding to immediately reassigned a newly free variable. This kind of list is most of the time uses with Tabu Search algorithm.

2.5 Extension Procedure

The choice of the next variable to instantiate obeys to two objectives. Firstly, the algorithm tries to reduce the explored width of the decision tree and secondly, to reduce the depth of the explored branches before the detection of the *deadend*. Two criteria are used to guide the decision of extension: a dynamic criterion determined by

the size of the filtered domains (called MRV for *Remaining Minimum Value*) and a static criterion referring to the number of constraints binding each variable. MRV is applied first and the variable degree is applied in case of equality on MRV between several variables. Then the choice of the value of the variable must obey to two rules. Firstly, the partial configuration resulting from the extension of the assignments with the decision ($x=d_x$) should not be declared as a *nogood* (the extended configuration is not in the permanent Tabu list). Secondly the value to be assigned should not be tabu at the current iteration (the extended configuration is not in the temporary Tabu list). Thus the candidate values must not be Tabu in any list. If whatever the value assigned to variable x, the extended configuration remains prohibited by the *nogoods* list, the part of the not extended configuration leading to this situation is added to the *nogoods* list. In that case the uninstantiation procedure (see section 3.4) is then called again to free a new variable. At last if a part of the values leads to a *nogood* and if the remainder values are Tabu within the temporary Tabu list, the uninstantiation procedure is also called again to work on another variable. In that case, we enforced the algorithm to diversify the search.

3 Tests and Results

3.1 Frequency Assignment Problem

The frequency assignment problem [AAR07][MAB02][GON07] is defined by a set of variables (transmitters, radio links, radio stations…) requiring each one a frequency channel. Each variable is defined by a domain of discrete subset of frequency channels. The variables of the problem are linked by electromagnetic compatibility constraints establishing the conditions of communication success. These constraints are often expressed as a minimal separation or the exact equality to respect by the frequency assignment. The theoretical reference problems are the colouring ones (k-colouring, T-colouring, etc). Within the framework of the CALMA project (*Combinatorial Algorithms for Military Applications*), several concurrent teams of European researchers worked on problem benchmarks. Eleven instances called SCEN were provided by the CELAR (*Centre Electronique de l'Armement, France*) and 14 others by a group of search of the University of Technology of Delft [BEN95]. The size of the problems varies between 200 to 1000 variables with domains of 44 values and the number of constraints is between 1134 and 5548.

In addition to the constraint satisfaction, some problems are described with an objective function on the spectrum usage. Two modes of optimization have been defined: the Min-Span mode that consists in reducing the spread of the frequency used by the variables (distance between the upper and the lower frequency used in the spectrum); and the Min-Order mode that consists in reducing the number of frequency used. The spectral optimization is carried out by the means of iterative re-start of constraint satisfaction procedure. In the case of Min-Span optimization, the algorithm is started with the original domains. If the algorithm finds a feasible solution, the higher frequency used in the solution d_{max} is removed from the domains and all the decisions $x=d_{max}$ in the feasible solution are cancelled. The partial solution thus generated is used as starting point for the search for a new feasible solution during the next

phase. The algorithm stops when the search for a feasible solution failed. The solution found during the preceding phase is then retained as optimal solution. In the case of Min-Order optimization, the same search process is followed, with the only difference that the least used frequency is eliminated after each phase.

3.2 Results and Comparisons

Several methods are proposed in the literature to solve the Frequency Assignment Problem. We present here the methods listed and compared on the research Web page of Zuse Institute of Berlin, ZIB, dedicated to CALMA, http://fap.zib.de/problems/CALMA. Two test sets are proposed in the Web site called CELAR [CELAR] and GRAPH [BEN95] and we work on those related to the Min-Span optimization to evaluate our new hybrid method. We like to emphasize that the results on other methods were not computed by us but by other teams. We give in Table 1 the results of the four best methods (among eight reported algorithms) for 4 first data set (it gives a general idea of performance of our approach), and in Table 2 the comparison of results on 12 data sets with CN-Tabu which is the most performing method on these benchmarks (it gives more precise output on our method performance). In Table 1 we put the 4 best results: a Branch-and-Cut algorithm proposed in [AHHJ96] uses a linear programming model, this algorithm uses problem specifications and generic techniques such as reduction methods, primal heuristics and branching rules to obtain optimal solutions; a Tabu Search algorithm proposed in [THL00]; a combination of the quadratic programming method followed by a rounding heuristic to obtain the final solution in [WTRJ97]; and an algorithm inspired by the Karmakar's interior point potential reduction approach where the method is applied on non-convex quadratic model of FAP [PAS98]. The columns indicate: the name of each instance, the best-known result (evaluated according to the highest frequency used in the spectrum) and the results of the different algorithms. The eight methods listed in ZIB web page find an optimal solution with a cost of 792 for the SCEN05 instance. But only the four methods presented above resolve the instances GRAPH03, GRAPH04 and GRAPH10. Among those 8 techniques there are only 3 methods solving optimally each problems. As well the four instances are solved in an optimal way by our method in less than 15 seconds. Our tests are carried out on PC Pentium IV, 2.4 GHz and 512MO of RAM. We have few information on the resolution context of the algorithms within the project CALMA. The data concerning the test machines, the time and the number of authorized retrials are unknown.

Then the Table 2 shows a comparison between the results of our method with those of the CN-Tabu method given in [DUP05]. The machine used by [DUP05] is a PC

Table 1. Comparison of our results with the 4 best methods for Min-Span

Scenario	Opt	Var	Cst	[AHHJ 96]	[THL00]	[WTRJ 97]	[PAS98]	Our results
SCEN 05	792	400	2598	792	792	792	792	792
GRAPH 03	380	200	1134	380	380	-	380	380
GRAPH 04	394	400	2244	394	394	-	394	394
GRAPH 10	394	680	3907	394	394	394	394	394

Pentium IV, 3 GHZ and 1 GB of RAM. CN-Tabu have proven its efficiency in the frame of ROADEF'01 challenge [Roadef01] by obtaining the first rank among several concurrent teams. The CN-Tabu method was applied on the instances presented in Table 2 with Min-Span objective. For each of the twelve instances, the table plots the name of the instance, the number of variables, the number of constraints to satisfy, the best result obtained by CN-Tabu (the highest frequency used in the spectrum), the CPU Time in seconds, and the success rate giving the percentage of optimal solutions got on 20 executions. The character "-" indicates the absence of this information.

Table 2. Comparison with the results of the CN-Tabu method for Min-Span

Scenario	Var	Cst	[DUP05]	Success rate	Time	Our results	Success rate	Time
SCEN 01	916	5548	680	-	1	680	100%	6
SCEN 02	200	1235	394	-	1	394	100%	1
SCEN 03	400	2760	666	-	1	**652**	**100%**	7
SCEN 05	400	2598	792	100%	1	792	100%	1
GRAPH 01	200	1134	408	-	1	408	100%	13
GRAPH 02	400	2245	394	-	1	394	100%	7
GRAPH 03	200	1134	380	100%	1	380	100%	5
GRAPH 04	400	2244	394	40%	1	**394**	**100%**	7
GRAPH 08	680	3757	**652**	**-**	15	680	100%	11
GRAPH 09	916	5246	**666**	**-**	664	694	100%	35
GRAPH 10	680	3907	394	25%	17	**394**	**100%**	15
GRAPH 14	916	4638	352	-	30	352	100%	22

The results presented in Table 2 show the competitiveness of our hybrid method. The grey cells indicate the best performing method in the basis of frequency used, then success rate, then computation time. It obtains a better result on SCEN03, GRAPH04, GRAPH10 and GRAPH14. However it remains less effective on GRAPH08 and GRAPH09 instances. On these two scenarios the frequency spectrum is organized in channels of width 14. The obtained solutions are then less good than the optimal solution by two channels. On GRAPH14 the performance is better on computational time knowing that the PC is less performing for our test (2,4GHz PIV instead of 3GHz PIV). Globally with the current results this new method is competitive with CN-Tabu which is the best one on these problems. One of the major contribution of our approach is in its deterministic behaviour; so when the optimal solution is found by the algorithm, it is always found that is with 100% success rate.

4 Conclusion

In this paper, we presented a promising hybrid method which takes advantages, on one hand, from the rigorous and optimal character of the exact approach and on the other hand, from the great flexibility of the heuristic approach. The principle of constraint propagation allows us to reduce search space and to record the blocking branches. It leads to a more effective exploration of the consistent solutions space.

The variables are ordered according to the conflicts degree and generated weights. These weights give the information about how and where operate the backtracking. Complementary to these mechanisms, we defined two Tabu lists to allow the algorithm to avoid search cycles and decision repetitions. The lists bring a kind of adaptive behaviour which traduces the information found by the algorithm during search. The result of the comparisons carried out showed that this new hybrid algorithm is a promising method which takes advantage from its deterministic behaviour. The effectiveness of the method depends on the relevance of the extension, uninstantiation and constraint propagation procedures. The improvement of these mechanisms should permit to get better performances of the method. Other extension mechanisms of the solution based on a prior study of the constraints graph structure are envisaged. This study concerns the identification of the difficult zones represented by the graph cliques. In a forthcoming work, our purpose is to study the application of the method to the frequency assignment problem with constraints on n-tuples which bind more than two variables.

References

[AAR07] Aardal, K.I., van Hoesel, S.P.M., Koster, A.M.C.A., Mannino, C., Sassano, A.: Models and solution techniques for frequency assignment problems (Updated version of a paper appeared in 4OR 1, 261-317, 2003) (January 2007)

[AHHJ96] Aardal, K.I., Hipolito, A., van Hoesel, C.P.M., Jansen, B.: A Branch-and-Cut Algorithm for the Frequency Assignment Problem. Research Memorandum 96/011, Maastricht University (1996)

[BAC95] Bacchus, F., van Run, P.: Dynamic variable reordering in CSPs. In: Montanari, U., Rossi, F. (eds.) CP 1995. LNCS, vol. 976, pp. 258–275. Springer, Heidelberg (1995)

[BEN95] van Benthem, H.P.: GRAPH Generating Radio Link Frequency Assignment Problems Heuristically. PhD thesis, Delft University of Technology (1995)

[CELAR] `http://www.inra.fr/internet/Departements/MIA/T//schiex/Doc/CELAR.shtml`

[CHE01] Chen, X., van Beek, P.: Conflict-Directed Backjumping Revisited. Journal of Artificial Intelligence Research 14, 53–81 (2001)

[DEB97] Debruyne, R., Bessière, C.: Some practicable filtering techniques for the constraint satisfaction problem. In: Proceedings of the International Joint Conference on Artificial Intelligence, Nagoya, Japan, pp. 412–417 (1997)

[DEB98] Debruyne, R.: Consistances locales pour les problèmes de satisfaction de contraintes de grande taille. PhD thesis, University Montpellier II (December 1998)

[DUP05] Dupont, A.: Étude d'une méta-heuristique hybride pour l'affectation de fréquences dans les réseaux tactiques évolutifs. PhD thesis, Université Montpellier II (October 2005)

[FOC02] Focacci, F., Laburthe, F., Lodi, A.: Local search and constraint programming. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics. Kluwer, Dordrecht (2002)

[GIN93] Ginsberg, M.: Dynamic backtracking. Journal of Artificial Intelligence Research 1, 25–46 (1993)

[GAR78] Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W.H. Freeman & Company, San Francisco (1978)

[GON07] Gondran, A., Baala, O., Caminada, A., Mabed, H.: Joint optimization of access point placement and frequency assignment in WLAN. In: 3rd IEEE/IFIP International conference in central Asia on Internet, September 26-28 (2007)

[HAO99] Hao, J.K., Galinier, P., Habib, M.: Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. Revue d'Intelligence Artificielle 13(2), 283–324 (1999)

[HAR80] Haralick, R.M., Elliott, G.L.: Increasing tree search efficiency for constraint satisfaction problems. Artificial Intelligence 14, 263–313 (1980)

[JAM03] Ben Jamaa, S., Altman, Z., Picard, J.M., Fourestié, B.: Optimisation de réseaux mobiles umts à l'aide des algorithmes génétiques. In: Dréo, J., Pétrowski, A., Siarry, P., Taillard, É.D. (eds.) Métaheuristiques pour l'optimisation difficile, coll. Algorithmes. Eyrolles (September 2003)

[JUS00] Jussien, N., Debruyne, R., Boizumault, P.: Maintaining Arc-Consistency within Dynamic Backtracking. In: Dechter, R. (ed.) CP 2000. LNCS, vol. 1894, pp. 249–261. Springer, Heidelberg (2000)

[JUS02] Jussien, N., Lhomme, O.: Local search with constraint propagation and conflict-based heuristics. Artificial Intelligence 139(1), 21–45 (2002)

[LAM06] Lambert, T.: Hybridation de méthodes complètes et incomplètes pour la résolution de CSP, October 2006. PhD thesis, Université de Nantes (2006)

[MAB02] Mabed, H., Caminada, A., Hao, J.K., Renaud, D.: A Dynamic Traffic Model for Frequency Assignment. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, Springer, Heidelberg (2002)

[PAS98] Pasechnik, D.V.: An Interior Point Approximation Algorithm for a Class of Combinatorial Optimization Problems: Implementation and Enhancements. Technical report, Delft University of Technology (1998)

[PAP94] Papadimitriou, C.H.: Computational Complexity. Addison-Wesley Publishing Company, Reading (1994)

[PRO93] Prosser, P.: Hybrid algorithms for the constraint satisfaction problem. Computational Intelligence 9(3), 268–299 (1993)

[Roadef01] `http://uma.ensta.fr/conf/roadef-2001-challenge/`

[SAB94] Sabin, D., Freuder, E.: Contradicting conventional wisdom in constraint satisfaction. In: Proc. of European Conference on Artificial Intelligence (ECAI 1994), Amsterdam, pp. 125–129 (1994)

[SHA98] Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M.J., Puget, J.-F. (eds.) CP 1998. LNCS, vol. 1520, pp. 417–431. Springer, Heidelberg (1998)

[SCH94] Schiex, T., Verfaillie, G.: Nogood recording for static and dynamic constraint satisfaction problems. International Journal on Artificial Intelligence Tools (IJAIT) (1994)

[THL00] Tiourine, S.R., Hurkens, C.A.J., Lenstra, J.K.: Local Search Algorithms for the Radio Link Frequency Assignment Problem. Telecommunication Systems 13, 293–314

[TSA93] Tsang, E.P.K.: Foundations of Constraint Satisfaction. Academic Press, London (1993)

[VAS00] Vasquez, M.: Résolution en variables 0-1 de problèmes combinatoires de grande taille par la méthode tabou. PhD thesis, université d'Angers, U.F.R. de Sciences (December 2000)

[WTRJ97] Warners, J.P., Terlaky, T., Roos, C., Jansen, B.: A Potential Reduction Approach to the Frequency Assignment Problem. Discrete Applied Mathematics 78, 251–282 (1997)

Evolutionary Optimisation of Kernel and Hyper-Parameters for SVM

Laura Dioşan[1,2], Alexandrina Rogozan[1], and Jean-Pierre Pécuchet[1]

[1] Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes, EA 4108
Institut National des Sciences Appliquées, Rouen, France
[2] Department of Computer Science, Faculty of Mathematics and Computer Science,
Babeş-Bolyai University, Cluj-Napoca, Romania
lauras@cs.ubbcluj.ro,
{arogozan,pecuchet}@insa-rouen.fr

Abstract. Support Vector Machines (SVMs) concern a new generation learning systems based on recent advances in statistical learning theory. A key problem of these methods is how to choose an optimal kernel and how to optimise its parameters. A (multiple) kernel adapted to the problem to be solved could improve the SVM performance. Therefore, our goal is to develop a model able to automatically generate a complex kernel combination (linear or non-linear, weighted or un-weighted, according to the data) and to optimise both the kernel parameters and SVM parameters by evolutionary means in a unified framework. Furthermore we try to analyse the architecture of such kernel of kernels (*KoK*). Numerical experiments show that the SVM algorithm, involving the evolutionary *KoK* performs statistically better than some well-known classic kernels and its architecture is adapted to each problem.

Keywords: Kernel of kernels, Support Vector Machines, Genetic Programming, Hyper-Parameters Optimization.

1 Introduction

The general problem of machine learning is to search a, usually very large, space of potential hypotheses to determine the one that will best fit the data. There are many learning algorithms today and their performances are related not only to the problem to be solved, but also to their parameters. Therefore, the best results can be achieved only by identifying the optimal values of these parameters. Although this is a very complex task, different optimisation methods have been developed in order to optimise the parameters of Machine Learning algorithms.

In this context, evolutionary computations have been theoretically and empirically proven robust for searching solutions in complex spaces and have been widely used in optimization, training neural networks, estimating parameters in system identification or adaptive control applications. Evolutionary algorithms form a subset of evolutionary computation in that they generally only involve techniques implementing mechanisms inspired by biological evolution such as reproduction, mutation, recombination, natural selection and survival of the fittest. Candidate solutions to the optimization problem

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 107–116, 2008.

play the role of individuals in a population, and the cost function determines the environment within which the solutions "live". Evolution of the population takes place after the repeated application of the above operators.

In 1995, Support Vector Machines (SVMs) marked the beginning of a new era in the paradigm of learning from examples. Rooted to the Statistical Learning Theory and the Structural Risk Minimization principle developed by Vladimir Vapnik at AT&T in 1963 [1], SVMs gained quickly attention from the Machine Learning community due to a number of theoretical and computational merits.

To date, various methods have been proposed to optimise the hyper-parameters of an SVM algorithm that uses a particular kernel. However, it was shown that any classical kernel achieve good enough performances for some classification problems [2]. A novel idea was to generate a new kernel function as a combination of classic kernels [3]. The combination which is obtained has to map the initial space into a larger one (the main purpose of a kernel function); in fact, this combination must be a more complex kernel. Therefore, we will denote it as a kernel of kernels (*KoK*). In this context, several questions arise concerning the architecture of a *KoK*: *Which are the kernels that have to be considered for the most efficient combination: different classic kernels and/or several instances of the same kernel, but with different parameters? How to optimise the hyper-parameters of a KoK-based SVM algorithm? Optimise each of the standard kernels and than involve them in the KoK or involve them in the KoK and than optimise?*

Trying to answer these questions, we will investigate in this paper the architecture of different *KoK*s and the importance of parameter optimisation in an SVM algorithm. We develop an evolutionary approach which is based on a previous model for designing *KoK*s [4]. The optimal expression of a *KoK* is actually found by involving a guided search process based on genetic operations: the selection has to provide high reproductive chances to the fittest *KoK*s, the crossover has to enable kernel-children to inherit quickly beneficial characteristics of their kernel-parents and the mutation has to ensure the diversity of the population and the exploration of the search space. After an iterative process, which runs more generations, an optimal evolutionary kernel of kernels (*eKoK*) and its parameters are provided. The design of these *KoK*s is analysed from two points of view: which are the most frequently involved standard kernels in the *KoK* and which are the parameters of these kernels.

The paper is organized as follows: Section 2 outlines the theory behind SVM classifiers giving a particular emphasis to the kernel functions. An overview of the related work in the field of SVM hyper-parameters optimisation is presented in Section 3. Section 4 describes the hybrid model utilised to optimise the *KoK* expression and its parameters. This is followed by Section 5 where the results of the experiments are presented and discussed. Finally, Section 6 concludes the paper.

2 Support Vector Machines

SVMs can solve binary or multiple-class problems. Originally, SVM have been proposed for solving binary classification problems [1]. Consequently, in what follows, the concepts within SVMs will be explained on binary-labelled data. However, our model can be applied for solving problems with any number of classes.

Suppose the training data has the following form: $D = (x_i, y_i)_{i=\overline{1,m}}$, where $x_i \in \Re^d$ represents an input vector and each y_i, $y_i \in \{-1, 1\}$, the output label associated to the item x_i. We are interested to find a function f that takes a set of unlabelled inputs x and provides the output $y = f(x)$ by using just the set of training observations D. This function can be viewed as a decision frontier (a hyper plane $\langle w, b\rangle$) that separates the input data in two regions: $f(x) = \langle w, x\rangle + b$,.

The value of each element of the weight vector w could be a measure of the relative importance of each of an item attributes for the classification of a sample. It has been shown that the optimal hyper-plane can be uniquely constructed by solving the following constrained quadratic optimisation problem:

$$\begin{array}{l} \text{minimise}_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \\ \text{subject to: } \; y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ \qquad\qquad\quad \xi_i \geq 0, \forall i \in \{1, 2, \ldots, m\}. \end{array} \tag{1}$$

Rather than solving this problem in its primal form it can be more easily solved in its dual formulation:

$$\begin{array}{l} \text{maximise}_{a \in \Re^m} \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i,j=1}^{m} a_i a_j y_i y_j \phi(x_i)\phi(x_j) \\ \text{subject to } \; \sum_{i=1}^{m} a_i y_i = 0, \\ \qquad\qquad\quad 0 \leq a_i \leq C, \forall i \in \{1, 2, \ldots, m\}. \end{array} \tag{2}$$

Instead of finding w and b, the goal now is to find the vector a and the bias value b, where each a_i represents the relative importance of a training sample x_i in the classification of a new sample. To classify a new sample, the quantity $f(x_s)$ is calculated as: $f(x_s) = \sum_i a_i y_i \langle x, x_i\rangle + b$ where b is chosen so that $y_i f(x_s) = 1$ for any i with $C > a_i > 0$. Then, a new sample x_s is classed as negative if $f(x_s) < 0$ and positive if $f(x_s) \geq 0$. Samples x_i for which the corresponding α_i are non-zero are known as *support vectors* since they lie closest to the separating hyperplane.

Because not all the input data-points are linear separable, it is suitable to use a kernel function. Cf. [5], a kernel is a function K, such that $K(x, z) = \langle \Phi(x), \Phi(z)\rangle$ for all $x, z \in X$, where Φ is a mapping from X to an (inner product) feature space $\mathcal{F}$. By using kernels, the solution could be expressed like an affine function: $f(x) = \langle w, \Phi(x)\rangle + b$, for some weight vector w. The kernel can be exploited whenever the weight vector can be expressed as a linear combination of the training points, $w = \sum_{i=1}^{m} a_i \Phi(x_i)$, implying that f can be expressed as: $f(x) = \sum_{i=1}^{m} a_i K(x_i, x) + b$. There are a wide choice for a positive definite and symmetric kernel K – see Table 1.

Table 1. The expression of several classic kernels

Name	**Expression**
Sigmoid	$K_{Sig}\ (x, z) = \tanh(\sigma x^T \cdot z + r)$
RBF	$K_{RBF}(x, z) = exp(-\sigma \lvert x - z\rvert^2)$
Liniar	$K_{Lin}\ (x, z) = x^T \cdot z$
Polynomial	$K_{Pol}\ (x, z) = (x^T \cdot z + coef)^d$

3 Related Work

While one of the first feelings about SVM algorithm is that it can solve a learning task automatically, it actually remains challenging to apply SVMs in a fully automatic manner. Questions regarding the choice of the kernel function and the hyper-parameters values remain largely empirical in the real-world applications. While default setting and parameters are generally useful as a starting point, major improvements can result from careful choosing of an optimal kernel.

Extensive explorations such as performing line search for one hyper-parameter or grid search for two hyper-parameters are frequently applied when such knowledge is unavailable [6]. More elaborated techniques for optimising the SVM hyper-parameters are the gradient-based approaches [2]. Keerthi et al. [7] have developed a hyper-parameter tuning approach based on minimizing a smooth performance validation function.

The previous approaches required to train the model several times with different hyper-parameter values. Therefore, new methods have been proposed to overcome these problems. Several promising recent approaches [8] are based on solution path algorithms, which can trace the entire solution path as a function of the hyper-parameters (the penalty error C and the kernel parameters) without having to train the model multiple times. A new study [9] has proposed to directly tackle the model selection by using out-of-sample testing as an optimization problem. EAs have optimised the hyper-parameters of an SVM classifier [10] as well.

Note that all the previous approaches deal only with a classic kernel, which is fixed *a priori*. Any kernel combination is considered in all these cases, because in the context of a *KoK* also the expression of such kernel combination must be optimised together with the hyper-parameters.

4 Evolutionary Kernel of Kernels (*eKoK*)

4.1 Model Architecture

This section describes an evolutionary approach for automatic design of *KoK*s and their parameter optimisation. This model is a hybrid one: it uses Genetic Programming (*GP*) [11] to construct positive and symmetric functions (*KoK*s), and optimizes the fitness function by using an SVM classifier (see Figure 1). A *GP* chromosome provides the analytic expression of such *KoK*. The model actually seeks to replace the expert domain knowledge concerning the design of the SVM's kernel function and the choice of its parameters, with a *GP* algorithm. The idea of combining more kernels by evolutionary means has been proposed in [4], but the purpose was only to generate a new complex kernel combination. The aim of the current work is to study the architecture of these *KoK*s and how they adapt to the problem to be solved.

The hybrid model we describe is structured on two levels: a macro level and a micro level. The macro level algorithm is a standard *GP* [11], which is used to evolve the mathematical expression of a *KoK*. The steady-state evolutionary model [12] is involved as an underlying mechanism for the *GP* implementation. A steady state algorithm is much more tolerant of poor offspring than a generational one. This is because in most implementations, the best individuals from a given generation will always be preserved in the

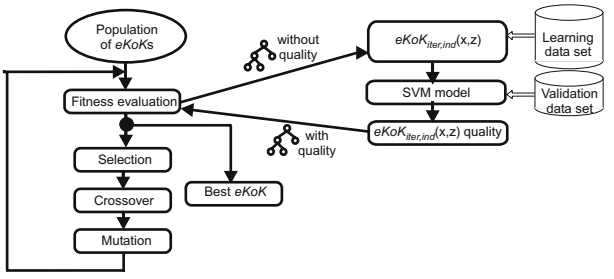


Fig. 1. Architecture of the hybrid model: a GP algorithm combined with an SVM one

next generation, giving themselves another opportunity to be selected for reproduction. The best individuals are therefore given more chances to pass on their successful traits. The *GP* algorithm starts by an initialisation step of creating a random population of individuals (seen as *KoK*s). The following steps are repeated until a given number of iterations is reached: two parents are selected using a binary selection procedure; the parents are recombined by using the sub-tree crossover [13] in order to obtain an offspring O; the offspring is than considered for mutation (a shrink mutation [14], followed by a grow mutation [14] are actually performed); the new individual O^* (obtained after mutation) replaces the worst individual W in the current population if O^* is better than W.

The micro level algorithm is an SVM classifier taken from *LIBSVM* [15] library. The original implementation of the SVM algorithm proposed in [15] allows using several well-known kernels (Linear, Polynomial, RBF and Sigmoid – see Table 1). In the numerical experiments, a modified version of this algorithm, which is based on the evolved *KoK* is also used. The quality of each *GP* individual is determined by running the SVM algorithm, which uses the *eKoK* encoded in the current chromosome. The accuracy rate estimated by the classifier (on the validation set) represents the fitness of the *GP* chromosome.

4.2 The Representation of the *eKoK*

The *GP* chromosome is a tree encoding the mathematical expression of a *KoK* and its parameters: the leaves contain either a classic parameterized kernel or an ephemeral random constant (viewed as a scaling or a shifting coefficient), while the internal nodes contain operations that preserve the key properties of a Mercer kernel ($+$, $\times$, exp). Moreover, the *GP* individual representation is constrained to satisfy the kernel algebra [5] (regarding the positiveness and the symmetry of the Gram matrix required by valid Mercer's kernels) – see [4] for more details.

Since our goal is to study the performance of various *KoK* architectures, three models are investigated. In the first model the *KoK* can be a combination of some standard kernels whose parameters have been optimised *apriori*. The other two models combines more kernels, but with different parameters. The GP algorithm will select the kernels of this set whose parameters are the best adapted to the problem to be solved. The second model combines more parameterised kernels of the same type, while the third model combines more parameterised kernels of different type. For all these models, two

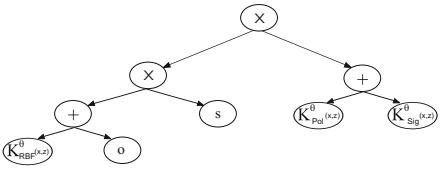


Fig. 2. A *GP* chromosome that encodes the expression of the following *KoK*: $(K^{\theta}_{RBF}(x,z) + o) \times s \times (K^{\theta}_{Pol}(x,z) + K^{\theta}_{Sig}(x,z))$

versions are investigated: a pure combination of kernels and a mixed one (kernels and weighting coefficients).

The well-known *grow method* [16], which is a recursive procedure, is used to initialize a *GP* individual. The root of each *GP* tree must be a function from FS. If a node contains a function, then its children are initialized either with another function or with a terminal (a kernel or a coefficient). The initialization process is stopped when is attained a leaf node or at the maximal depth of the tree (the nodes from the last level will be initialised by terminals). The maximal depth of a *GP* chromosome has to be large enough in order to assure a sufficient search space for the optimal expression of our evolutionary *KoK*. An example of a *GP* chromosome is depicted in Figure 2.

4.3 Fitness Assignment

The evaluation of the chromosome quality is based on a cross-validation process. Therefore, some information about the data set partitioning must be provided before to describe the fitness assignment process. The data sample was randomly divided into two sets: a training set (80%) - for model building - and a testing set (20%) - for performance assignment. The training set was than randomly partitioned into learning (2/3) and validation (1/3) parts.

The SVM model based on the *eKoK* that is encoded in the current *GP* tree uses the learning subset for training the SVM model and the validation subset for classification performance assignment. The quality of an *eKoK* can be measured by the classification accuracy rate estimated on the validation data set: the number of correctly classified items over the total number of items belonging to the validation set. Note that we deal with a maximization problem: the greater accuracy rate, the better *eKoK* is. Once the *GP* iterations end, the optimal *eKoK*, which corresponds to the best *GP* chromosome is utilised by SVM algorithm in order to classify the test items.

4.4 Genetic Operations

Selection. The selection operator chooses from the current population which individuals will act like parents in order to create the next generation. Selection has to provide high reproductive chances to the fittest individuals but, at the same time, it has to preserve the exploration of the search space.

Crossover. The crossover operator assures the diversity of the *KoK*s and is performed in a tree-structure preserving way in order to ensure the validity of the offspring. The

proposed model uses the standard cutting-point crossover [11] with the particularity that the offspring has to contain at least one kernel in its leaves. This crossover type has been used because it is able to guarantee a quite quickly convergence of the *GP* algorithm.

Mutation. The purpose of the mutation operator is to create new individuals by small and stochastic perturbations of a chromosome. For a *GP*-based *KoK*, a cutting point is randomly chosen: the sub-tree belonging to that point is deleted and a new sub-tree is grown there by applying the same random growth process that was used to generate the initial population. Note that the maximal depth allowed for the *GP* trees limits the growth process.

The initialization, recombination and mutation operators always generate valid *KoK*s.

5 Experimental Validation and Discussions

This section reports on the experimental validation of *eKoK* on a standard set of benchmark problems [17]. The hybrid model is based on TinyGP [18] framework of *GP* algorithm and LIBSVM [15] framework of SVM classifier. The performances of *eKoK*s are evaluated on several binary classification problems taken from Machine Learning Repository UCI and Statlog database: $P_1(34, 351)$[1] – *ionosphere*, $P_2(10, 683)$ – *breast*, $P_3(13, 270)$ – *heart*, $P_4(123, 4217)$ – *a1a*, $P_5(123, 2591)$ – *a2a*.

A population of 50 individuals is evolved during 50 iterations, which are reasonable limits to assure the diversity and convergence of our *eKoK*s. The maximal depth of a *GP* tree is limited to 10 levels, which allows encoding till $(2^{10} - 1)!$ combinations of kernels and coefficients. This maximal depth was fixed by tacking into account the bloat problem (the uncontrolled growth of programs during *GP* runs without (significant) return in terms of fitness [18]). Furthermore, several empirical tests indicated that the efficient kernel-trees do not expand to more than 10 levels. The crossover and mutation operations are performed with 0.8 and 0.3, respectively, probabilities, values that are generally recommended in the specialised literature.

The selection of the kernel parameters has the same importance as the optimisation of the kernel expression. In order to determine good values of these parameters, it is important to search on the right scale. The default value for the C parameter is that suggested in [2]: $s^2 = \frac{1}{m}\sum_{i=1}^{m} KM_{i,i} - \frac{1}{m^2}\sum_{i=1}^{m}\left(\sum_{j=1}^{m} KM_{ij}\right)$ from an $m \times m$ kernel matrix KM. This value is actually used in our numerical experiments performed in order to evolve the expression of a *KoK* function.

Several kernels of kernels are evolved in this experiment by using different terminal sets (TSs). The purpose of this experiment is to emphasise the contribution of the parameter optimisation to the classification performance. These terminal sets can contain only several standard kernels *KTS* or also some coefficients $MTS = KTS \cup \{o, s\}$. Note that in our experiments, these constants could be either scaling or shifting coefficients from $[0, 1]$ range. The *eKoK*s based on *KTS* are un-weighted combinations (pure combinations of kernels), while the *eKoK*s based on *MTS* could be weighted combinations.

[1] The number of characteristics and the number of items are given for each problem.

Therefore, the *TS*s actually used in the numerical experiments are:

1. a *TS* composed by different standard kernels (see Table 1) with fixed parameters – $KTS_1 = \{K^{\theta}_{Lin}, K^{\theta}_{Pol}, K^{\theta}_{RBF}, K^{\theta}_{Sig}\}$ where the parameters θ of each simple kernel have been optimised by the parallel grid search method (for each data set) *apriori* – before to involve them in the *eKoK*.
2. a *TS* that contains more kernels of the same type, but with different parameters; we select the well-known *RBF* kernel: $KTS_2 = \{K^{\theta}_{RBF}\}$, where $\theta = \sigma_{qt} = q \cdot 10^t$, $q = \overline{1,9}$, $t = \overline{-5,0}$. The difference between these *RBF* kernels is determined by the bandwidth value (the σ parameter);
3. a TS composed by different standard kernels, each of them with different parameters $KTS_3 = \{K^{\theta}_{Pol}, K^{\theta}_{RBF}, K^{\theta}_{Sig}\}$ where the parameters θ of each standard kernel have been considered in some discrete ranges: for the degree d of the Polynomial kernel 15 values (from 1 to 15) are considered, for the bandwidth σ of the RBF kernel the following values: $\sigma_{qt} = q \cdot 10^t$, $q = \overline{1,9}$, $t = \overline{-5,0}$ are considered and for the Sigmoid kernel all the combination between σ_{qt} and r, where $r = 10^u$, $u \in \{-1, 0, 1\}$ are taken into account;
4. a *TS* composed by different standard kernels with fixed parameters (*apriori* optimised) and coefficients $MTS_1 = KTS_1 \cup \{o, s\}$;
5. a *TS* that contains RBF kernel, but with different values for the σ parameter and coefficients $MTS_2 = KTS_2 \cup \{o, s\}$;
6. a TS of different standard kernels with different parameters and coefficients $MTS_3 = KTS_3 \cup \{o, s\}$.

The results found in literature indicate that these discrete spaces of parameters are the most suitable for an efficient classification. The improvement obtained by using a finer discretisation of the parameter space or a continuous space is no relevant (by tacking into account the computational effort that must be performed). Furthermore, the guided search (based on the efficiency of an *eKoK*) involved by the evolutionary algorithm is able to detect in the discrete space the optimal values of these parameters (and implicit the corresponding kernels).

The performances of the *eKoK*s based on various *TS*s and their confidence intervals are presented in Table 2: the first six rows contain the accuracy rates (for each problem) estimated by the SVM algorithm involving our evolutionary *KoK*s on the test set (unseen data). Table 2 also presents the performances of three classic kernels for all the test problems (the last three rows).

The values from Table 2 allows realising several comparisons:

- *standard kernels vs. combined kernels* – for each problem, *eKoK*s perform better than the standard kernels;
- *weighted vs. un-weighted KoKs* – for P_3 and P_4 problems a pure combination of kernels (un-weighted *eKoK* based on *KTS*) performs better than a mixed (kernels and coefficients) combination (weighted *eKoK* based on *MTS*), while for the other problems the weighted *eKoK*s achieve to the best classification performances;
- *fixed vs. optimised parameters* – the parameter optimisation seems to be very important: for the un-weighted *eKoK*s this optimisation determines the performance improvement for 3 problems (out of 5), while for the weighted *eKoK*s it improves the classification quality for all the problems.

Table 2. The accuracy rate of various kernels and their confidence intervals estimated on the test data

		P_1	P_2	P_3	P_4	P_5
eKoK	KTS_1	88.89±1.03	97.81±0.13	86.92±0.51	85.88±0.14	87.01±0.19
	KTS_2	86.11±1.13	97.81±0.13	87.57±0.50	84.26±0.14	88.74±0.18
	KTS_3	86.11±1.13	97.81±0.13	86.98±0.51	84.27±0.14	86.93±0.19
	MTS_1	86.11±1.13	97.81±0.13	86.98±0.51	84.18±0.14	87.44±0.19
	MTS_2	88.89±1.03	98.03±0.13	87.57±0.50	84.38±0.14	89.08±0.18
	MTS_3	91.67±0.90	98.03±0.13	86.98±0.51	84.38±0.14	88.99±0.18
Standard kernels	K_{Pol}	77.77±1.36	97.58±0.14	85.79±0.53	84.26±0.14	86.24±0.20
	K_{RBF}	80.55±1.29	97.81±0.13	85.21±0.54	83.65±0.14	83.49±0.21
	K_{Sig}	66.67±1.54	97.81±0.13	77.91±0.63	82.73±0.14	84.52±0.21

In addition, we have investigated which are the most often involved kernels in an evolutionary *KoK*. The expression of the *KoK*s from the last generation are inspected and the results are presented in Table 3 for two problems.

We can observe that for each problem and for each possible terminal set different results are obtained indicating the importance of the adaptation of the evolved *KoK* and its parameters to the problem that must be solved.

As a general conclusion, we can affirm that the *eKoK*s has to be adapted to the problem and its characteristics. By tacking into account the values from Tables 2 and 3 we cannot identify an *eKoK* (based on a particular TS) which works better than the other ones for all the problems. Furthermore, the parameter optimisation has a strong influence on SVM performances. However, it is not enough to evolve a new kernel. To obtain good generalisation, it is also necessary to optimise the hyper-parameters. Their values could affect the quality of the SVM solution.

Table 3. The most used kernels (and their parameters) contained by the evolved *KoK*

KoK	KTS_1	KTS_2	KTS_3	MTS_1	MTS_2	MTS_3
P_1	K_{Lin}	$K_{RBF}^{\sigma=0.08}$	$K_{Sig}^{\sigma=0.07,r=-0.1}$	K_{RBF}	$K_{RBF}^{\sigma=0.01}$	$K_{Sig}^{\sigma=0.08,r=-0.1}$
P_3	K_{RBF}	$K_{RBF}^{\sigma=0.02}$	$K_{Pol}^{d=2}$	K_{Lin}	$K_{RBF}^{\sigma=0.03}$	$K_{Sig}^{\sigma=0.08,r=-0.1}$

6 Conclusions

A hybrid model to optimise SVM kernel and its parameters has been studied. Several numerical experiments have been performed to emphasize the importance of parameter optimisation and of kernel adaptation.

We will focus our further work on the validation of *eKoK* model developed in this paper for large data sets and using multiple data sets for the training stage; this could help to evolve kernels that are more generic. Furthermore, we plan to evolve *KoK*s for

feature selection tasks and to use them in order to solve classification problems with heterogeneous data. In this way, we should favour the data fusion process.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
2. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for Support Vector Machines. Machine Learning 46(1/3), 131–159 (2002)
3. Lanckriet, G.R.G., et al.: Learning the kernel matrix with Semidefinite Programming. Journal of Machine Learning Research 5, 27–72 (2004)
4. Diosan, L., Rogozan, A., Pécuchet, J.P.: Optimising multiple kernels for SVM by Genetic Programming. In: van Hemert, J., Cotta, C. (eds.) EvoCOP 2008. LNCS, vol. 4972, pp. 230–241. Springer, Heidelberg (2008)
5. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
6. Staelin, C.: Parameter selection for Support Vector Machines. Technical Report HPL-2002-354R1, Hewlett Packard Laboratories (2003)
7. Keerthi, S., Sindhwani, V., Chapelle, O.: An efficient method for gradient-based adaptation of hyperparameters in SVM models. In: NIPS 2006, pp. 1–10. IEEE Computer Society, Los Alamitos (2006)
8. Bach, F.R., Thibaux, R., Jordan, M.I.: Computing regularization paths for learning multiple kernels. In: NIPS, pp. 1–10 (2004)
9. Bennett, K., Hu, J., Ji, X., Kunapuli, G., Pang, J.S.: Model selection via bilevel optimization. In: IJCNN 2006, pp. 1922–1929. IEEE Computer Society, Los Alamitos (2006)
10. Friedrichs, F., Igel, C.: Evolutionary tuning of multiple SVM parameters. Neurocomputing 64, 107–117 (2005)
11. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
12. Syswerda, G.: A study of reproduction in generational and steady state Genetic Algorithms. In: Rawlins, G.J.E. (ed.) FOGA, pp. 94–101. Morgan Kaufmann, San Francisco (1991)
13. Koza, J.R.: Genetic programming II. MIT Press, Cambridge (1994)
14. Angeline, P.J.: Two self-adaptive crossover operators for genetic programming. In: Angeline, P.J., Kinnear Jr., K.E. (eds.) Advances in Genetic Programming 2, pp. 89–110. MIT Press, Cambridge (1996)
15. Chang, C.C., Lin, C.J.: LIBSVM: a library for Support Vector Machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
16. Banzhaf, W.: Genetic programming: An introduction: On the automatic evolution of computer programs and its applications (1998)
17. Newman, D.J., Hettich, S., C.B., Merz, C.: UCI repository of ML databases (1998)
18. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming (2008)

Lot-Sizing and Sequencing on a Single Imperfect Machine

Alexandre Dolgui[1], Mikhail Y. Kovalyov[1,2], and Kseniya Shchamialiova[1]

[1] Industrial Engineering and Computer Science Centre (G2I),
Ecole des Mines de Saint Etienne,
158, cours Fauriel, 42023 Saint Etienne Cedex 2, France
dolgui@emse.fr, shchamialiova@emse.fr
http://www.emse.fr/∼dolgui
[2] Faculty of Economics, Belarusian State University, and United Institute
of Informatics Problems, National Academy of Sciences of Belarus,
Nezavisimosti 4, 220030 Minsk, Belarus
koval@newman.bas-net.by

Abstract. We study a problem of lot-sizing and sequencing several discrete products on a single machine. A sequence dependent setup time is required between the lots of different products. The machine is imperfect in the sense that it can produce defective items, and furthermore breakdown. The number of the defective items for each product is given as an integer valued non-decreasing function of the manufactured quantity for this product. The total machine breakdown time is given as a real valued non-decreasing function of the manufactured quantities of all the products. The objective is to minimize the total cost of the demand dissatisfaction, provided that a given upper bound on the completion time for the last item has been satisfied.

Keywords: Lot-sizing, Sequencing, Imperfect production.

1 Introduction

A single facility (machine) is used to manufacture items of n discrete products in lots. A lot is the maximal set of items of the same product, which are manufactured with no inserted item of another product. Each lot is preceded by a sequence dependent setup time. The size of a lot is the number of its items. The machine is imperfect in the sense that it can produce defective items, and furthermore, it can breakdown. Defective items cannot be repaired. Therefore, they are disposed of. No item can be manufactured during the setup or machine breakdown times. Moreover, setting up is impossible during the breakdown time. The following parameters are given for each product i:

b_i - a demand for the good quality items (counted in the number of items), $b_i > 0$;
c_i - a per unit cost for the unsatisfied demand, $c_i > 0$;
t_i - a processing requirement for every single item, $t_i > 0$;

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 117–125, 2008.

$s_{i,j}$ $(s_{j,i})$ - a setup time required to switch from processing of a lot of the product i (j) to a lot of the product j (i), $s_{i,j} \geq 0$, $s_{j,i} \geq 0$;
$s_{0,i}$ - a setup time required to start processing of a lot of the product i, if it is processed first on the machine, $s_{0,i} \geq 0$;
$f_i(x)$ - a non-decreasing integer valued function representing the number of the defective items, if the total number of the manufactured items of product i is equal to x, $f_i(0) = 0$, and $f_i(x) < x$ for $x = 1, 2, \ldots$

We assume that the setup times satisfy the *triangle inequality* such that $s_{i,j} + s_{j,k} \geq s_{i,k}$ for $i = 0, 1, \ldots, n$, $j = 1, \ldots, n$, $k = 1, \ldots, n$. The total machine breakdown time before the last item has been produced is determined by a given non-negative real valued function $T(x_1, \ldots, x_n)$ non-decreasing in each argument, where x_i is the total number of the manufactured items (both defective items and good quality items) of the product i, $i = 1 \ldots, n$. Functions $f_i(x)$, $i = 1 \ldots, n$, and $T(x_1, \ldots, x_n)$ of an adequate type can be obtained by a statistical analysis of historical data concerning the machine.

The decision variables are the product lots (their sizes) and their sequence. The objective of our problem, denoted as P-Cost, is to minimize the total linear cost of the demand dissatisfaction, $\sum_{j=1}^{n} c_i \max\{0, b_i - (x_i - f_i(x_i))\}$, subject to $C_{\max} \leq T_0$, where $C_{\max}$ is the completion time of the last item, provided that all the product demands are satisfied, and T_0 is a given upper bound on the completion time of the last item.

Problem P-Cost lies at the intersection of the two research fields: scheduling with batching and lot-sizing, and optimal lot-sizing for imperfect production systems. The majority of the problems in the former field are deterministic and discrete, and those in the latter field are mainly stochastic and continuous (exceptions can be found, for example, in Inderfurth et al. [11] and Inderfurth et al. [12]). Surveys on scheduling with batching and lot-sizing are given by Potts and Van Wassenhove [16], Potts and Kovalyov [15] and Allahverdi et al. [1]. Lot-sizing models for imperfect production systems were studied by Rosenblatt and Lee [17], Groenevelt, Pintelon and Seidmann [9], Flapper et al. [10], Chiu, Ting and Chiu [4], Buscher and Lindner [2], to name a few.

The most similar problem to P-Cost was studied by Dolgui, Levin and Louly [6]. The difference is that the quantities of the defective items and the machine breakdown times were assumed to be random variables with the given probabilities and distribution functions, and the objective was to maximize the probability of the demand satisfaction within a given production time period. Dolgui, Levin and Louly [6] presented a three-level decomposition approach to solve their problem. We will suggest a more efficient combination of the optimization and approximation techniques to solve problem P-Cost and some of its special cases. This new problem possesses the following properties.

Property 1. There exists an optimal solution of the problem P-Cost, in which items of the same product are manufactured in at most one lot.

An item shifting technique can be used to prove this property. Note: the triangle inequality is needed for the correct implementation of this technique.

Property 2. There exists an optimal solution of the problem P-Cost,whose sequence of (non-empty) lots minimizes the total setup time.

This property can be proved by noting that a reduction in the total setup time does not increase the $C_{\max}$ value, nor does it increase the dissatisfaction of any demand.

One can see that there exists an optimal solution of the problem P-Cost, which satisfies both Property 1 and Property 2. Property 1 was implicitly used by Dolgui, Levin and Louly [6] in their previous work. Property 2 was explicitly discussed and used by them for an optimal sequence, which includes a single lot of each product (formulation of their problem suggests that at least one lot must be created for each product). Hereafter, we shall consider only solutions satisfying Properties 1 and 2.

Problem P-Cost is studied in Section 2. We show that optimal sequencing and lot-sizing decisions can be separated for this problem. Furthermore, a set of products for which at least one item is manufactured has to be determined. A dynamic programming algorithm is described, which determines optimal sequences for all possible sets of products. We further prove that the lot-sizing subproblem of P-Cost is NP-hard even in the "fraction defective" case. We also present a fully polynomial time approximation scheme for this case. The paper concludes with a summary of the results.

Note: another new problem exists which we denote as problem P-Time that we are also working on [7]. For this the objective is to minimize the completion time for the last item, $C_{\max}$, provided that all the product demands are satisfied. While not in the scope of this article, in our oral presentation at the conference, we will also discuss a model for the problem P-Time.

2 Problem P-Cost

In problem P-Cost, the demands for the good quality items are not required to be satisfied, but total linear demand dissatisfaction cost should be minimized, subject to $C_{\max} \leq T_0$, where T_0 is a given upper bound on the completion time for the last item. In an optimal solution of this problem, it may happen that no item of a product is manufactured. Therefore, a *selection* decision has to be made which determines the set of products with at least one manufactured item.

Optimal product permutations for all possible selection decisions can be found in $O(n^2 2^n)$ time by the following dynamic programming algorithm, which is similar to the well-known algorithm of Held and Karp [13] developed for the TSP with triangle inequality. In this algorithm, values $T(S, i)$ are recursively computed, where $T(S, i)$ is the minimum total setup time for processing a set of products S, provided that product $i \in S$ is processed last. The initialization is $T(S, i) = s_{0,i}$ for $S = \{i\}$, $i = 1, \ldots, n$, and the recursion for $S \subseteq \{1, \ldots, n\}$, $|S| = 2, 3, \ldots, n$, is given by

$$T(S, i) = \min_{j \in S \setminus \{i\}} \{T(S \setminus \{i\}, j) + s_{j,i}\}. \tag{1}$$

For any set S, the minimum total setup time $T^*(S)$ can be calculated as $T^*(S) = \min_{i \in S}\{T(S,i)\}$ in $O(|S|)$ time. All the relevant values $T^*(S)$ can be computed in $O(n^2 2^n)$ time. Given S and $T^*(S)$, the corresponding optimal permutation $\pi^*(S)$ can be found in $O(n)$ time by backtracking the dynamic programming algorithm described above.

In the rest of this section, we assume that the optimal selection and sequencing decisions have been made: products of a set N have been selected for manufacturing and their optimal permutation has been found. To facilitate our presentation, let $N = \{1, \ldots, n\}$. Let x_i denote the size of the only lot of product i, $i = 1, \ldots, n$, and let $x = (x_1, \ldots, x_n)$. Problem P-Cost reduces to the following lot-sizing problem, which we denote as P-Cost-Sizes.

$$\text{Minimize } D(x) := \sum_{i=1}^{n} c_i \max\{0, b_i - (x_i - f_i(x_i))\},$$

$$\text{subject to } E(x) := \sum_{i=1}^{n} t_i x_i + T(x) \leq T_1, \quad x_i \in Z_+, \quad i = 1, \ldots, n, \quad (2)$$

where T_1 is equal to T_0 minus the corresponding optimal total setup time. Let $x^* = (x_1^*, \ldots, x_n^*)$ be an optimal solution of this problem.

Observe that if $n = 1$ and the only function $f_1(x)$ is represented by an oracle, then the question whether there exists a solution of problem P-Cost-Size such that $D(x) \leq 0$ is equivalent to the NP-hard problem P formulated in the next Theorem:

Theorem 1. *Let $f(x)$ be a non-decreasing integer valued function, which is given by an oracle, such that $f(x) < x$ for $x = 1, 2, \ldots$ The problem of deciding whether $x - f(x) \geq b$ for $x \in Z_+$, which we refer to as problem P, is NP-hard.*

The proof of the Theorem is in line with the proof from Cheng and Kovalyov [3]. Therefore, the following statement holds.

Statement 1. *Problem P-Cost-Sizes is NP-hard even if $n = 1$ and the only function $f_1(x)$ is represented by an oracle.*

For specific functions $f_i(x)$, $i = 1, \ldots, n$, problem P-Cost-Sizes might be polynomially solvable. However, we shall now prove that it is NP-hard even in the "fraction defective" case, where all the functions $f_i(x)$, $i = 1, \ldots, n$, are rounded linear functions. Recall that the problem P-Time-Sizes is solvable in $O(n)$ in this case [7]. Then we shall present an efficient $(1 + \varepsilon)$-approximation algorithm for the "fraction defective" case of the problem P-Cost-Sizes.

Theorem 2. *Problem P-Cost-Sizes is NP-hard even if $f_i(x) = \lfloor \frac{x}{3} \rfloor$, $b_i = 2$, $i = 1, \ldots, n$, and $T(x) = 0$.*

Proof. We shall use a reduction from the NP-complete PARTITION problem, see Garey and Johnson [8]: Given positive integer numbers $a_1, \ldots, a_m$ and A, where $\sum_{i=1}^{m} a_i = 2A$, is there a subset $X \subset M := \{1, \ldots, m\}$ such that $\sum_{i \in X} a_i = A$?

Given any instance of PARTITION, we construct the following instance of the problem P-Cost-Sizes. Set $n = m$, $T(x) = 0$, $T_1 = 3A$, $f_i(x) = \lfloor \frac{x}{3} \rfloor$, $b_i = 2$, $c_i = t_i = a_i$, $i = 1, \ldots, n$. We show that PARTITION has a solution if and only if there exists a solution x to the constructed instance of the problem P-Cost-Sizes such that $D(x) \leq A$. We call such a solution a *feasible solution.*

Part "only if". Assume that set X is a solution to PARTITION. Construct a vector $x = (x_1, \ldots, x_n)$ such that $x_i = 2$ if $i \in X$, $x_i = 1$ if $i \notin X$. We have

$$D(x) = \sum_{i=1}^{m} a_i \max\left\{0, 2 - \left(x_i - \left\lfloor \frac{x_i}{3} \right\rfloor\right)\right\} = \\ = \sum_{i=1}^{m} a_i \max\{0, 2 - x_i\} = \sum_{i=1}^{m} a_i(2 - x_i) = A, \quad (3)$$

and $E(x) = \sum_{i=1}^{m} a_i x_i = 3A = T_1$, i.e., a feasible solution to the constructed instance of the problem P-Cost-Sizes exists.

Part "if". Let there exist a feasible solution for the constructed instance of the problem P-Cost-Sizes. Given such a solution, introduce sets $X_1 = \{i \mid x_i = 1\}$ and $X_2 = \{i \mid x_i \geq 2\}$. Since $\max\{0, 2 - (x_i - \lfloor \frac{x_i}{3} \rfloor)\} = 0$ if $x_i \in \{2, 3, \ldots\}$, we must have $D(x) = \sum_{i \in X_1} a_i \leq A$ and $E(x) = \sum_{i \in X_1} a_i + 2 \sum_{i \in X_2} a_i \leq 3A$. Taking into account $X_1 \cup X_2 = M$, we deduce that $E(x) = \sum_{i \in X_1} a_i + 2(2A - \sum_{i \in X_1} a_i) = 4A - \sum_{i \in X_1} a_i \leq 3A$, which together with $D(x) = \sum_{i \in X_1} a_i \leq A$ implies $\sum_{i \in X_1} a_i = A$, i.e., set $X := X_1$ is a solution to problem PARTITION. □

We shall now present a *Fully Polynomial Time Approximation Scheme (FPTAS)* for the "fraction defective" case of the problem P-Cost-Sizes, which is hereby denoted as P-Cost-Sizes-FD. In this case, $f_i(x) = \lfloor \alpha_i x \rfloor$, and $T(x) = \sum_{i=1}^{n} \gamma_i t_i x_i$, where α_i and γ_i are rational numbers such that $0 \leq \alpha_i < 1$ and $0 \leq \gamma_i < 1$, $i = 1, \ldots, n$. Let L denote the largest denominator in the irreducible fractions representing the numbers α_i and γ_i, $i = 1, \ldots, n$. A family $\{A_\varepsilon\}$ of $(1 + \varepsilon)$-approximation algorithms A_ε constitutes an FPTAS for the problem P-Cost-Sizes-FD if the running time of each algorithm A_ε is bounded by a polynomial of n, $\log p_{\max}$, and $1/\varepsilon$, where $p_{\max} = \max\{\max_{1 \leq i \leq n}\{c_i, b_i, t_i\}, L\}$ is the maximum numerical parameter.

Problem P-Cost-Sizes-FD can be formulated as follows.

$$\text{Minimize} D(x) := \sum_{i=1}^{n} c_i \max\{0, b_i - (x_i - \lfloor \alpha_i x_i \rfloor)\}, \\ \text{subject to } E(x) := \sum_{i=1}^{n} (1 + \gamma_i) t_i x_i \leq T_1, x_i \in Z_+, \imath = 1, \ldots, n. \quad (4)$$

Let $x^* = (x_1^*, \ldots, x_n^*)$ denote its optimal solution. If $E(x^{(1)}) > T_1$, where $x^{(1)} = (1, ..., 1)$, then problem P-Cost-Sizes-FD has no feasible solution. Assume without loss of generality that $E(x^{(1)}) \leq T_1$. For our FPTAS, we shall need lower and upper bounds V and U such that $0 < V \leq D(x^*) \leq U$. An upper bound can be determined as $U = D(x)$, where x is an arbitrary feasible solution, for

example, $U = D(x^{(1)}) = \sum_{i=1}^{n} c_i(b_i - 1)$, where $x^{(1)} = (1, \ldots, 1)$. We shall now show how a positive lower bound can be found. Denote

$$h_i = \min\{x \mid x \in Z_+, b_i - x + \lfloor \alpha_i x \rfloor \le 0\} = \\ = \begin{cases} \frac{b_i - 1}{1 - \alpha_i} + 1, & \text{if } \frac{b_i - 1}{1 - \alpha_i} \text{ is integer,} \\ \lceil \frac{b_i - 1}{1 - \alpha_i} \rceil, & \text{if } \frac{b_i - 1}{1 - \alpha_i} \text{ is not integer,} \end{cases} \quad i = 1, \ldots, n. \quad (5)$$

If $E(x^{(h)}) \le T_1$, where $x^{(h)} = (h_1, \ldots, h_n)$, then $x^* = x^{(h)}$ because $D(x^{(h)}) = 0$. If $E(x^{(h)}) > T_1$, then for each feasible solution $x = (x_1, \ldots, x_n)$, there exists an index i such that $b_i - x + \lfloor \alpha_i x \rfloor \ge 1$, which implies

$$D(x^*) \ge \min_{1 \le i \le n} \{c_i\} := V \ge 1. \quad (6)$$

Now assume without loss of generality that lower and upper bounds are known such that $0 < V \le D(x^*) \le U$. We have shown that these values V and U can be computed in $O(n)$ time. Determine a *scaling parameter* $\delta = \frac{\varepsilon V}{n}$ and formulate the following "rounded problem", denoted as P-Rou.

$$\text{Minimize } w(x) := \sum_{i=1}^{n} \left\lfloor \frac{v_i(x_i)}{\delta} \right\rfloor,$$

$$\text{subject to } E(x) \le T_1, \quad \text{and}$$

$$x_i \in \{r_i(0), r_i(1), \ldots, r_i(\left\lfloor \frac{U}{\delta} \right\rfloor)\}, i = 1, \ldots, n, \quad (7)$$

where

$$v_i(x_i) = c_i \max\{0, b_i - (x_i - \lfloor \alpha_i x_i \rfloor)\},$$

$$r_i(l) = \min\{x \mid x \in Z_+, \left\lfloor \frac{v_i(x)}{\delta} \right\rfloor \le l\}, l = 0, 1, \ldots, \left\lfloor \frac{U}{\delta} \right\rfloor. \quad (8)$$

The inequality $\left\lfloor \frac{v_i(x)}{\delta} \right\rfloor \le l$ is equivalent to $b_i - x_i + \lfloor \alpha_i x_i \rfloor < \frac{(l+1)\delta}{c_i}$. Since the left hand side of this inequality is integer, it is in turn equivalent to $b_i - x_i + \lfloor \alpha_i x_i \rfloor < \left\lceil \frac{(l+1)\delta}{c_i} \right\rceil$ and further to $\left\lceil \frac{(l+1)\delta}{c_i} \right\rceil + x_i - b_i > \alpha_i x_i$, which can be written as

$$x_i > \frac{b_i - \lceil \frac{(l+1)\delta}{c_i} \rceil}{1 - \alpha_i} := h_i^{(l)}. \quad (9)$$

Thus,

$$r_i(l) = \begin{cases} \max\{1, h_i^{(l)} + 1\}\}, & \text{if } h_i^{(l)} \text{ is integer,} \\ \max\{1, \lceil h_i^{(l)} \rceil\}\}, & \text{if } h_i^{(l)} \text{ is not integer,} \end{cases}$$

$$l = 0, 1, \ldots, \left\lfloor \frac{U}{\delta} \right\rfloor, \ i = 1, \ldots, n. \quad (10)$$

Problem P-Rou can be formulated in $O(n\left\lfloor \frac{U}{\delta} \right\rfloor) = O(n^2 \frac{U}{V})$ time. Let x^0 denote an optimal solution of this problem. Similar to Kovalyov [14], it can be easily proved that any exact algorithm for the problem P-Rou is an $(1+\varepsilon)$-approximation algorithm for the problem P-Cost-Sizes-FD. Let $E_j(f)$ denote the minimum value of the function $\sum_{i=1}^{n}(1+\gamma_i)t_i x_i$ on the set of vectors $x = (x_1, \ldots, x_j)$ such that $\sum_{i=1}^{j} \left\lfloor \frac{v_i(x_i)}{\delta} \right\rfloor = f$. Problem P-Rou can be solved by the following dynamic programming algorithm.

Algorithm A_ε (FPTAS for problem P-Cost-Sizes-FD)

Step 1. (Initialization) Set $E_j(f) = 0$ if $f = 0$, $j = 0$, and $E_j(f) = \infty$, otherwise. Set $j = 1$.

Step 2. (Recursion) For $f = 0, 1, \ldots, \left\lfloor \frac{U}{\delta} \right\rfloor$, compute the following:

$$E_j(f) = \min\left\{E_{j-1}\left(f - \left\lfloor \frac{v_j(x_j)}{\delta} \right\rfloor\right) + (1+\gamma_j)t_j x_j \mid x_j \in \{r_j(0), \ldots, r_j(f)\}\right\}. \tag{11}$$

If $j < n$, then set $j = j + 1$ and repeat Step 2; otherwise, go to Step 3.

Step 3. (Optimal solution) Compute the optimal solution value

$$w(x^0) = \min\left\{f \mid E_n(f) \le T_1, f = 0, 1, \ldots, \left\lfloor \frac{U}{\delta} \right\rfloor\right\} \tag{12}$$

and find the corresponding optimal solution x^0 by backtracking.

The running time of algorithm A_ε is $O\left(\frac{n^3}{\varepsilon^2}\left(\frac{U}{V}\right)^2\right)$. By applying the Bound Improvement Procedure in Tanaev, Kovalyov and Shafransky [18] (see English translation in Chubanov, Kovalyov and Pesch [5]), value D^0 can be found in $O\left(n^3 \log\log \frac{U}{V}\right)$ time such that $0 < D^0 \le D(x^*) \le 3D^0$. Then we can set $V = D^0$ and $U = 3D^0$. The family of algorithms $\{A_\varepsilon\}$ with the Bound Improvement Procedure incorporated in it constitutes an FPTAS for the problem P-Cost-Sizes-FD, and the following statement holds.

Statement 2. *There exists an FPTAS for the problem P-Cost-Sizes-FD with the running time* $O\left(\frac{n^3}{\varepsilon^2} + n^3 \log\log(\sum_{i=1}^{n} c_i b_i)\right)$ *of each algorithm.*

3 Conclusions

Deterministic problem P-Cost of optimal sequencing and lot-sizing of several products on a single imperfect machine have been studied. We have shown that optimal sequencing and lot-sizing decisions can be made separately for this problem. The lot-sizing decision was called P-Cost-Sizes problem.

The problem of optimal P-Cost-Sizes was proved NP-hard even if a function representing the number of defective items is given by an oracle. The P-Cost-Sizes was also proved NP-hard in the "fraction defective" case for which an FPTAS with running time $O\left(\frac{n^3}{\varepsilon^2} + n^3 \log\log(\sum_{i=1}^{n} c_i b_i)\right)$ was developed.

Computer experiments on classes of real-life and randomly generated instances made in order to verify the applicability of the suggested approaches were accomplished. The numerical results prove the efficiency of the proposed models.

Acknowledgment. The research of Kovalyov M.Y. was partially supported by the Programm of Fundamental and Applied Research of the Republic of Belarus, project "Mathematical Models 28".

References

1. Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y.: A survey of scheduling problems with setup times or costs. European Journal of Operational Research (in press) Available online 13 November 2006
2. Buscher, U., Lindner, G.: Optimizing a production system with rework and equal sized batch shipments. Computers & Operations Research 34, 515–535 (2007)
3. Cheng, T.C.E., Kovalyov, M.Y.: An unconstrained optimization problem is NP-hard given an oracle representation of its objective function: a technical note. Computers & Operations Research 29, 2087–2091 (2002)
4. Chiu, S.W., Ting, C.-K., Chiu, Y.-S.P.: Optimal production lot sizing with rework, scrap rate, and service level constraint. Mathematical and Computer Modelling 46, 535–549 (2007)
5. Chubanov, S., Kovalyov, M.Y., Pesch, E.: An FPTAS for a single-item capacitated economic lotsizing problem. Mathematical Programming, Ser. A 106, 453–466 (2006)
6. Dolgui, A., Levin, G., Louly, M.-A.: Decomposition approach for a problem of lotsizing and sequencing under uncertainties. International Journal of Computer Integrated Manufacturing 18, 376–385 (2005)
7. Dolgui, A., Kovalyov, M.: Multi-product lot-sizing on an imperfect single machine, Research Report G2I 2007-500-009, Ecole des Mines de Saint Etienne, France, 16 pages (2007)
8. Garey, M.R., Johnson, D.S.: Computers and intractability: a guide to the theory of NP-completeness. W.H. Freeman and Co., San Francisco (1979)
9. Groenevelt, H., Pintelon, L., Seidmann, A.: Production lot sizing with machine breakdowns. Management Science 38, 104–120 (1992)
10. Flapper, S.D.P., Fransoo, J.C., Broekmeulen, R.A.C.M., Inderfurth, K.: Planning and control of rework in the process industries: a review. Production Planning & Control 1, 26–34 (2002)
11. Inderfurth, K., Janiak, A., Kovalyov, M.Y., Werner, F.: Batching work and rework processes with limited deterioration of reworkables. Computers and Operations Research 33, 1595–1605 (2006)
12. Inderfurth, K., Kovalyov, M.Y., Ng, C.T., Werner, F.: Cost minimizing scheduling of work and rework processes on a single facility under deterioration of reworkables. International Journal of Production Economics 105, 345–356 (2007)
13. Held, M., Karp, R.M.: A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied Mathematics 10, 196–210 (1962)
14. Kovalyov, M.Y.: A rounding technique to construct approximation algorithms for knapsack and partition type problems. Applied Mathematics and Computer Science 6, 101–113 (1996)

15. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: A review. European Journal of Operational Research 120, 228–249 (2000)
16. Potts, C.N., Van Wassenhove, L.N.: Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. Journal of the Operational Research Society 43, 395–406 (1992)
17. Rosenblatt, M.J., Lee, H.L.: Economic production cycles with imperfect production processes. IIE Transactions 18, 48–55 (1986)
18. Tanaev, V.S., Kovalyov, M.Y., Shafransky, Y.M.: Scheduling Theory. Group Technologies, Minsk, IEC NANB (in Russian) (1998)

Best and Worst Optimum for Linear Programs with Interval Right Hand Sides

Virginie Gabrel, Cecile Murat, and Nabila Remli

University Paris Dauphine, LAMSADE
Place du Maréchal de Lattre de Tassigny,
F-75775 Paris Cedex 16, France
Virginie.Gabrel@lamsade.dauphine.fr,
murat@lamsade.dauphine.fr,
nabila.remli@lamsade.dauphine.fr

1 Problem Statement

In optimization, it is used to deal with uncertain and inaccurate factors which make difficult the assignment of a single plausible value to each model parameters. Two approaches are possible: in the first one, a single nominal value is assigned to each parameter, the corresponding optimal solution is computed, then the interval in which each parameter can vary in order to preserve optimality solution is determined; the second approach consists in taking into account in the model to optimize, the possible variations of each parameter. In mathematical programming, the first approach is known as sensitivity analysis (see e.g. [6]). For the second approach, stochastic optimization may be applied for some problems in which parameters value can be described by probability laws (see for example [4]). When it is not possible nor relevant to associate probability laws to parameters, another way amounts to assign a set of possible values to each parameter. Two models may be considered: in the first one, a finite set of values is assigned to each uncertain model coefficient; in the second one, each uncertain model coefficient is associated with an interval number. In this paper, we only consider this second model called interval linear programming.

The choice of one value in each interval corresponds to a scenario. The induced robust optimization problem is to determine a single solution which is optimal for all scenarios. In general, such a solution does not exist and the problem is to determine a "relatively good" solution for all scenarios (see for example [2,8,10]). When uncertainty concerns feasible solution set, robustness problems have been less studied (see for example [9,7]). Nevertheless, a lot of real optimization problems include uncertainty and inaccuracy factors on feasible solutions set. For example, when a linear program represents a production problem in which the right hand sides equal to some forecast demands on several periods, it may be much more relevant to replace each right hand side coefficient by a suitable interval number.

In this paper, we consider general linear programs in which each right hand side b_i is an interval number $[\underline{b_i}, \overline{b_i}]$. It is assumed that each b_i can take on any

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 126–134, 2008.

value from the corresponding interval regardless of the values taken by other coefficients.

The aim of this work is to define the theoretical complexity of the best and worst optimum problems (firstly introduced by Chinneck and Ramadan in [5]). In the first part, we consider linear programs with inequality constraints. In the second part, we deal with those containing equality constraints, and then, we extend the results to general linear programs. In each case, we characterize optimal solutions.

2 Uncertainty on Right Hand Sides: Main Results

When uncertainty (represented by interval numbers) concerns right hand side constraints, only few results have already been obtained. The difficulty comes from the fact that the set of feasible solutions is not exactly known. Thus, any solution may not be feasible for all interval right hand side.

In [5], Chinneck and Ramadan consider general linear programs with interval coefficients (simultaneously in objective function, matrix constraints and right hand sides). The goal is to compute the best possible optimum and the worst one over all possible scenarios in order to provide a kind of robustness information: "The range of the objective function between the best and the worst optimum values gives a sense of the risk involved... For example, the specific values of the uncertain coefficients can be chosen to reflect a conservative or a risk-taking strategy."

In [5], algorithms are proposed to determine best and worst optimum but none complexity result is given. They consider separately linear problems with variables restricted in sign and equality or inequality constraints. They propose polynomial time algorithms for determining the best optimum of a linear program with variables restricted in sign, and the worst optimum of linear program with inequality constraints and variables restricted in sign. They define an exponential time algorithm for computing the worst optimum of a linear program with equality constraints and variables restricted in sign. Moreover, the authors remark that the complexity of the algorithm grows when variables are not restricted in sign.

When only right hand sides are interval numbers in a linear program, we show in this paper that only two cases have to be distinguished for complexity analysis. Firstly, we consider the easier case of linear programs with general inequality constraints (whatever the sign of each variable is), and secondly, we study the much more difficult case of linear programs with equality constraints. In each case, we characterize optimal solutions.

3 Linear Programs with Interval Right Hand Sides: The Case of Inequality Constraints

We consider the following linear program with n variables and m constraints

$$(P^b_{\geq}) \begin{cases} \min cx \\ s.t\ Ax \geq b \end{cases}$$

We suppose that each b_i varies in the interval $[\underline{b_i}, \overline{b_i}]$. For all $b \in [\underline{b}, \overline{b}]$, we denote $X^b_{\geq}$ the polyhedron defined by $\{x \in \mathbb{R}^n : Ax \geq b\}$ and we suppose that $X^b_{\geq}$ is a nonempty bounded polyhedron. As usual, we denote in this paper $v(P)$ the optimal solution value of the optimization problem (P).

3.1 Best Optimal Solution

Our objective is to determine the minimum value of the optimal solution of $(P^b_{\geq})$ when b varies in the interval $[\underline{b}, \overline{b}]$. The best optimal solution problem can be written as follows

$$(B_{\geq}) \begin{cases} \min v(P^b_{\geq}) \\ s.t\ \underline{b} \leq b \leq \overline{b} \end{cases}$$

Theorem 1. *$(B_{\geq})$ can be solved in polynomial time.*

Proof. It is sufficient to remark that $(B_{\geq})$ is equivalent to the following linear program

$$\begin{cases} \min cx \\ s.t\ Ax \geq b \\ \quad \underline{b} \leq b \leq \overline{b} \end{cases}$$

Moreover, it is possible to characterize the scenario which gives the best optimal solution. Since $X^b_{\geq} \subseteq X^{\underline{b}}_{\geq}$ for all b in $[\underline{b}, \overline{b}]$, we have $v(P^b_{\geq}) \geq v(P^{\underline{b}}_{\geq})$ and consequently $v(B_{\geq}) = v(P^{\underline{b}}_{\geq})$.

3.2 Worst Optimal Solution

Our objective is to determine the maximum value of the optimal solution of $(P^b_{\geq})$ when b varies in the interval $[\underline{b}, \overline{b}]$. The worst optimal solution problem can be written as follows

$$(W_{\geq}) \begin{cases} \max v(P^b_{\geq}) \\ s.t\ \underline{b} \leq b \leq \overline{b} \end{cases}$$

We have $X^{\overline{b}}_{\geq} \subseteq X^b_{\geq}$ for all b in $[\underline{b}, \overline{b}]$. Thus $v(P^b_{\geq}) \leq v(P^{\overline{b}}_{\geq})$ and the theorem 2 follows.

Theorem 2. *$(W_{\geq})$ can be solved in polynomial time since $v(W_{\geq}) = v(P^{\overline{b}}_{\geq})$.*

4 Linear Programs with Interval Right Hand Sides: The Case of Equality Constraints

In this section, we consider the following linear program with n variables and m equality constraints

$$(P^b_{=}) \begin{cases} \min cx \\ s.t\ Ax = b \\ \quad x \geq 0 \end{cases}$$

We suppose that each b_i varies independently in the interval $[\underline{b_i}, \overline{b_i}]$. For all $b \in [\underline{b}, \overline{b}]$, we denote $X^b_{=}$ the polyhedron defined by $\{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\}$ and we suppose that $X^b_{=}$ is a nonempty polyhedron. Two cases must be distinguished:

- $n = m$ and the rank of matrix A equals to n. In this case, the problem $(P^b_=)$ has only one feasible solution.
- $n > m$ and the rank of matrix A equals to m. In this case, $(X^b_=)$ is unbounded and we suppose that $(P^b_=)$ presents a finite optimal solution for all b.

We introduce two sets $\overline{X} = \bigcup_{b\in[\underline{b},\overline{b}]} X^b_=$ and $\underline{X} = \bigcap_{b\in[\underline{b},\overline{b}]} X^b_=$. Given a solution $x \in \overline{X}$ and a scenario b, we have:

- either x belongs to $X^b_=$ and its value is equal to cx,
- or x does not belong to $X^b_=$ and, by convention, we set its value to $+\infty$.

Let us remark that $(P^b_=)$ can be formulated as a $(P^b_\geq)$ problem as follows

$$\begin{cases} \min cx \\ s.t\ Ax \geq b \\ \quad -Ax \geq -b \\ \quad \underline{b} \leq b \leq \overline{b} \end{cases}$$

But for such a problem, each right hand side does not vary independently to each other. Each b_i appears twice with opposite sign in two different constraints. Thus we have to study specifically the case of equality constraints.

4.1 Best Optimal Solution

The best optimal solution problem is equivalent to

$$(B_=) \begin{cases} \min v(P^b_=) \\ s.t\ \underline{b} \leq b \leq \overline{b} \end{cases}$$

Theorem 1. *The problem $(B_=)$ can be solved in polynomial time.*

The proof is equivalent to the proof 3.1 given in the case of linear program with inequality constraints.

Another formulation of $(B_=)$ can be obtained by introducing additional variables noted by $z \in \mathbb{R}^m$. For $i = 1, \ldots, m$, each z_i variable, defined in $[0, 1]$, represents the deviation from the lower bound $\underline{b_i}$ in the interval $[\underline{b_i}, \overline{b_i}]$ and we have

$$\forall b_i \in [\underline{b_i}, \overline{b_i}], b_i = \underline{b_i} + z_i(\overline{b_i} - \underline{b_i}) \text{ with } z_i \in [0, 1]$$

So, $(B_=)$ can be written

$$\begin{cases} \min cx \\ s.t\ Ax = \underline{b} + z(\overline{b} - \underline{b}) \\ \quad x \geq 0 \\ \quad 0 \leq z \leq 1 \end{cases}$$

Let us remark that this reformulation, with z_i variables, is inspired by the robustness approach proposed by Bertsimas and Sim [3].

With this formulation, one may characterize the scenario which leads to the best optimal solution.

Theorem 2. *The best optimal solution can be obtained with an extreme scenario, that is to say, $\forall i = 1, \ldots, m$, z_i equals to 1 or 0.*

4.2 Worst Optimal Solution

The problem of determining the worst optimal solution can be formulated as follows

$$(W_=) \begin{cases} \max v(P^b_=) \\ \quad s.t\ \underline{b} \leq b \leq \overline{b} \end{cases}$$

In order to analyze the theoretical complexity of $(W_=)$, we deal separately with the simplest case in which $n = m$ and $rank(A) = n$ and the more difficult case in which $n > m$ and $rank(A) = m$.

Case $n = m$ and $rank(A) = n$

Theorem 3. *When $n = m$ and $rank(A) = n$, the problem $(W_=)$ is solvable in polynomial time.*

Proof 1. *We remark that, when the problem has a unique feasible solution, the optimal solution of $(P^b_=)$ is equal to $x^* = A^{-1}b$. $(W_=)$ is equivalent to the following linear program:*

$$\begin{cases} \max cA^{-1}b \\ \quad s.t\ A^{-1}b \geq 0 \\ \qquad \underline{b} \leq b \leq \overline{b} \end{cases}$$

Case $n > m$ and $rank(A) = m$

At first, we have to prove the following lemma:

Lemma 1. *The following quadratic program is NP-hard:*

$$(Q_{\mathrm{X}}) \begin{cases} \max cx \\ \quad s.t\ x \in X \\ \qquad \underline{c} \leq c \leq \overline{c} \end{cases}$$

with $X \subseteq \mathbb{R}^n$ nonempty bounded polyhedron.

Proof 2. *Let us consider the following linear program with a bounded feasible solutions set and interval coefficients in the objective function*

$$\begin{cases} \max cu \\ \quad s.t\ \Delta u \leq \beta \end{cases}$$

with $c \in [\underline{c}, \overline{c}]$, $u, c \in \mathbb{R}^n$, $\beta \in \mathbb{R}^m$ and $\Delta \in \mathbb{R}^{m \times n}$. Averbakh and Lebedev prove in [1] that the problem of computing the maximum regret value of a given u is NP-hard. This problem can be written as follows

$$f_{\mathrm{REG}}(u) = \max_{\substack{\underline{c} \leq c \leq \overline{c} \\ \Delta v \leq \beta}} \{c(v - u)\}$$

By setting $x = v - u$, we obtain

$$f_{\text{REG}}(u) = \begin{cases} \max cx \\ \quad s.t\ \Delta x \leq \beta - \Delta u \\ \qquad \underline{c} \leq c \leq \overline{c} \end{cases}$$

If we denote the bounded set $X = \{x \in \mathbb{R}^n : \Delta x \leq \beta - \Delta u\}$ we have

$$f_{\text{REG}}(u) = \begin{cases} \max cx \\ \quad s.t\ x \in X \\ \qquad \underline{c} \leq c \leq \overline{c} \end{cases}$$

Thus, (Q_{X}) has the same complexity as $f_{\text{REG}}(u)$ which is strongly NP-hard and the lemma is proven.

Now, we prove that (Q_{X}) remains NP-hard even if the feasible solution set is unbounded.

Lemma 2. *The following quadratic problem is NP-hard:*

$$(Q_{\text{X}'}) \begin{cases} \max cx \\ \quad s.t\ x \in X' \\ \qquad \underline{c} \leq c \leq \overline{c} \end{cases}$$

with $X' \subseteq \mathbb{R}^n$ a nonempty unbounded polyhedron.

Proof 3. *The problem (Q_{X}) with $X \subseteq \mathbb{R}^{n-1}$ is equivalent to*

$$(Q_{\text{X}'}) \begin{cases} \max \sum_{j=1}^{n-1} c_j x_j - c_n x'_n \\ \quad s.t\ x \in X \\ \qquad x'_n \geq 0 \\ \qquad \underline{c} \leq c \leq \overline{c} \\ \qquad 0 \leq c_n \leq M \end{cases}$$

It is to be noted that the polyhedron $X' = \{x \in X, x'_n \geq 0\}$ is a nonempty unbounded polyhedron (since X is nonempty).

The optimal solution of $(Q_{\text{X}'})$ is $(x_1^*, \ldots, x_{n-1}^*, 0)$ with $(x_1^*, \ldots, x_{n-1}^*)$ being the optimal solution of (Q_{X}). Thus $(Q_{\text{X}'})$ is NP-hard.

Theorem 4. *When $n > m$, $rank(A) = m$ and $(P_=^b)$ has finite optimal solution for all $b \in [\underline{b}, \overline{b}]$, the problem $(W_=)$ is NP-hard.*

Proof 4. *For a given b, the dual program of $(P_=^b)$, is*

$$(D_=^b) \begin{cases} \max b^t \lambda \\ \quad s.t\ A^t \lambda \leq c^t \end{cases}$$

where $\lambda = (\lambda_i)_{i=1,\ldots,m}$ and λ_i is the dual variable of the i^{th} constraint $\sum_{j=1}^{n} a_{ij} x_j = b_i$.

According to the strong duality theorem, one can replace $v(P^b_=)$ *by* $v(D^b_=)$ *in* $(W_=)$ *as follows*

$$v(W_=) = \max_{\underline{b} \leq b \leq \overline{b}} \max_{A^t\lambda \leq c^t} b^t\lambda$$

which is equivalent to the following quadratic program

$$(Q) \begin{cases} \max b^t\lambda \\ s.t\ A^t\lambda \leq c^t \\ \underline{b} \leq b \leq \overline{b} \end{cases}$$

According to lemmas 1 and 2 problems $(W_=)$ *and* (Q) *are NP-hard.*

Moreover, the scenario which leads to a worst optimal solution is an extreme scenario. Considering (Q), one can remark that for a given feasible solution λ, the b_i variables can be separately optimized since $\max_{\underline{b} \leq b \leq \overline{b}} \sum_{i=1}^m b_i\lambda_i = \sum_{i=1}^m \max_{\underline{b}_i \leq b_i \leq \overline{b}_i} b_i\lambda_i$. Thus, for all $i = 1, \ldots, m$, if $\lambda_i \geq 0$ then $b_i^* = \overline{b}_i$ otherwise, if $\lambda_i < 0$ then $b_i^* = \underline{b}_i$. Chinneck and Ramadan in [5] observe also that extreme scenarios are those of interest to determine a worst optimum and they give an exact algorithm which enumerates the 2^m extreme scenarios.

5 Linear Programs with Interval Right Hand Sides: The General Case

In this section, we consider a general linear program with n variables, m_1 equality constraints and m_2 inequality constraints

$$(P^{b,b'}) \begin{cases} \min cx \\ s.t\ Ax = b \\ A'x \geq b' \end{cases}$$

We suppose that each b_i (resp. b'_i) varies in the interval $[\underline{b_i}, \overline{b_i}]$ (resp. $[\underline{b'_i}, \overline{b'_i}]$). For a fixed b and b', we denote $X^{b,b'}$ the polyhedron defined by $\{x \in \mathbb{R}^n : Ax = b,\ A'x \geq b'\}$ and we suppose that $X^{b,b'}$ is a nonempty polyhedron.

5.1 Best Optimal Solution

The best optimal solution problem is equivalent to

$$(B) \begin{cases} \min v(P^{b,b'}) \\ s.t\ \underline{b} \leq b \leq \overline{b} \\ \underline{b}' \leq b' \leq \overline{b}' \end{cases}$$

Theorem 5. (B) *can be solved in polynomial time.*

Proof 5. *The proof is the same as the proof 3.1. It leads to solve the linear program*

$$(B) \begin{cases} \min cx \\ s.t\ Ax = b \\ A'x \geq \underline{b}' \\ \underline{b} \leq b \leq \overline{b} \end{cases}$$

Table 1. Main results

	best opt	worst opt
$\begin{cases} \min cx \\ s.t\ Ax \geq b \end{cases}$ with $\underline{b} \leq b \leq \overline{b}$	polynomial	polynomial
case $n = m$ and $rank(A) = n$ $\begin{cases} \min cx \\ s.t\ Ax = b \\ x \geq 0 \end{cases}$ with $\underline{b} \leq b \leq \overline{b}$	polynomial	polynomial
case $n > m$ and $rank(A) = m$ $\begin{cases} \min cx \\ s.t\ Ax = b \\ x \geq 0 \end{cases}$ with $\underline{b} \leq b \leq \overline{b}$	polynomial	NP-hard

5.2 Worst Optimal Solution

The problem of determining the worst optimal solution can be formulated as follows

$$(W) \begin{cases} \max v(P^{b,b'}) \\ s.t\ \underline{b} \leq b \leq \overline{b} \\ \quad \underline{b}' \leq b' \leq \overline{b}' \end{cases}$$

Case $n = m_1$ and $rank(A) = n$.

Theorem 6. *When $n = m_1$ and $rank(A) = n$, the problem (W) is solvable in polynomial time.*

Proof 6. *(W) can be written as a linear program as follows:*

$$\begin{cases} \max cA^{-1}b \\ s.t\ A'A^{-1}b \geq \overline{b}' \\ \quad \underline{b} \leq b \leq \overline{b} \end{cases}$$

Case $n > m_1$ and $rank(A) = m_1$.

Theorem 7. *When $n > m_1$ and $rank(A) = m_1$, the problem (W) is NP-hard.*

The proof is equivalent to proof 4.

6 Conclusion

In this article, we study the theoretical complexity of the best (worst) optimum problem for linear program with interval right hand sides. In the following table, the main results are summarized:

The best and worst optimum values can be seen as indicators for dealing with uncertainty in a decision problem. In fact, the best (worst) solution is optimal only for a particular scenario and their performance on the other scenarios is unknown (and can be far away from optimality). For evaluating the robustness of a solution, another approach must be applied. A classical one comes from decision theory and amounts to apply some suitable criteria, like the worst case criteria or the maximum regret criteria. It will be very interesting to analyze the relationship between the best and worst optimum problems and those of determining the optimal solutions according to the best and worst case criterion. This will be the subject of future research.

References

1. Averbakh, I., Lebedev, V.: On the complexity of minmax regret linear programming. European Journal of Operational Research 160, 227–231 (2005)
2. Ben-Tal, A., Nemirovski, A.: Robust solutions of uncertain linear programs. Operations Research Letters 25, 1–13 (1999)
3. Bertsimas, D., Sim, M.: The price of robustness. Operations Research 52(1), 35–53 (2004)
4. Birge, J., Louveaux, F.: Introduction to stochastic programming. Springer, Berlin (1997)
5. Chinneck, J.W., Ramadan, K.: Linear programming with interval coefficients. The Journal of the Operational Research Society 51(2), 209–220 (2000)
6. Chvatal, V.: Linear programming. W.H. Freeman, New York (1983)
7. Minoux, M.: Duality, robustness, and 2-stage robust lp decision models. Technical Report 7, Robustness in OR-DA, Annales du LAMSADE (2007)
8. Mulvey, J.M., Vanderbei, R.J., Zenios, S.A.: Robust optimization of large-scale systems. Operations Research 43(2), 264–281 (1995)
9. Soyster, A.L.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research 21, 1154–1157 (1973)
10. Vincke, P.: Robust solutions and methods in decision-aid. Journal of multi-criteria decision analysis 8, 181–187 (1999)

Transit Network Re-timetabling and Vehicle Scheduling

Valérie Guihaire[1,2] and Jin-Kao Hao[2]

[1] PERINFO, 41 avenue Jean Jaurès, 67100 Strasbourg, France
vguihaire@perinfo.com
[2] LERIA, Université d'Angers, 2 boulevard Lavoisier, 49045 Angers, France
hao@info.univ-angers.fr

Abstract. In the transit planning literature, network timetabling and vehicle scheduling are usually treated in a sequential manner. In this paper, we focus on combining important features of these two steps, and underline how their simultaneous optimization is meaningful and can bring important improvements to both quality of service and level of resources required. We deal with the objectives of networkwide quality of service through number and quality of the transfers and evenness of the line headways, and with the resources side through number of vehicles needed. Our approach is adapted to the problem faced by regulating authorities, treating among others intermodality, multi-periods for headways and travel times, and complex timetable schemes. We introduce an optimization procedure based on Iterated Local Search and present computational experiments carried out on a real large transit network, showing substantial improvements in both quality of service and level of resources compared to the current practice.

Keywords: Mass transit, Timetabling, Scheduling, Transfer Synchronization, Iterated Local Search.

1 Introduction

Transit network timetabling is the step of transit planning during which the quality of service of the offer is determined and the level of resources needed is strongly influenced (lower-bounded). In general, the offer is defined first, to create transfer possibilities and respect headway bounds. Vehicle allocation only occurs afterwards, thoroughly constraining the problem of minimizing the number of buses needed. Despite the strong relation between these two problems, most studies focus on a single side, due to the intrinsic complexity of each of them.

Depending on the focus of the problem treated, denominations for transit network timetabling include Schedule Synchronization [6], Transfer Time Optimization [3] or Transfer Coordination [12]. Objectives assigned to these problems include minimizing total waiting time, minimizing transfer waiting time or maximizing the number of simultaneous arrivals [9]. This problem has often been modeled as a Quadratic Semi-Assignment Problem (QSAP) [6,1] which aims

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 135–144, 2008.

at minimizing the global transfer waiting time of passengers in the network by setting the first departure time of each line. However, QSAP cannot capture some important operational constraints such as variable headways. The author of [11] proposed a constructive heuristic to set line runs departure times. The objective was to minimize the total transfer waiting time while allowing variable headways between runs. However, the evenness of headways was not considered as an objective, neglecting an important factor in service quality. In many studies, headway evenness is rather computed as a function of initial waiting time weighted by the number of users waiting [7]. This implies availability of the desired boarding times of the passengers or the assumption of arrival time distribution functions.

In order to include additional degrees of freedom, the authors of [4] proposed a Non-Linear Mixed Integer Problem model in which stopping times are allowed to vary. However, due to the level of complexity involved, this study considered a single transfer node. In [7], the authors used a Genetic Algorithm on a networkwide basis. A single common period is considered for the whole network, meaning fixed running times and unvarying headway demand for each route.

Vehicle scheduling, on the other hand, consists in assigning vehicles to line runs and depots, thereby creating the so-called vehicle services. Several aspects have been studied, considering different levels of complexity, such as number of depots or fleet homogeneity/heterogeneity [5]. In the case of regulating authorities, information regarding depots and fleet are unavailable.

So far, the simultaneous approach of optimal fleet distribution and timetabling of transit systems has only been superficially explored. It has often been narrowed to integrating constraints on the number of available vehicles in the timetabling problem and considering schedules without interlining. The first study that we are aware of that considers the number of vehicles as an objective of the transit network timetabling problem is reported in [4]. The authors proposed a genetic algorithm to tackle a combination of transfer coordination and vehicle scheduling problems. However, since the representation is cumbersome, the restrictive case of a single transfer stop with multiple lines is studied.

In this paper, we consider the combination of transit network timetabling with vehicle scheduling from the point of view of regulating authorities, meaning the consideration of depots is not needed yet. Given a pre-defined lines network; the current timetable; groups of lines sharing resources; running times; headway periods; and levels of importance of the transfers; the goal is to define a synchronized network timetable and the associated vehicle assignment with respect to a set of constraints and objectives. Our approach is based on three different levels of evaluation: headway evenness is calculated per line, level of resources is computed per group of lines, and transfer optimization works at the network level. We present a solution method combining both perspectives, leading to a very flexible decision-aid tool. An Iterated Local Search procedure is developed, which is based on the exploration of two types of neighborhoods aiming at alternatively intensifying and diversifying the search.

2 Problem Description

Our problem consists in assigning a departure time and a vehicle to each line run in the network, with respect to a set of objectives and constraints. In this part, we detail the characteristics, variables and domains, inputs, constraints and objectives of our approach. Let us state a few definitions first:

- A *route* is a sequence of stops, and a line is a route with a direction. In the rest of this paper, we will consider only lines.
- A *line run* is a trip on the line, characterized by a departure time.
- An *external line* is any activity connected in time and space with the transit network (e.g. a train line, a factory quitting). This information is used to create intermodal transfers.
- The *turnaround time* is the time needed by a vehicle at the end of a line run to get ready for the next trip and possibly catch up with some lateness accumulated with respect to the planned schedule.
- The *headway* of a line is the time separating the service of its main stop by consecutive runs. It is the inverse of the frequency over a time period.
- A *network timetable* is composed of line timetables, which in turn correspond to the set of all arrival and departure times for the stops served by each run.
- A *vehicle assignment* is the entire sequence of line runs assigned to a vehicle.

2.1 Properties of Our Approach

Our model includes a set of interesting properties that render it particularly flexible, realistic and adapted to planners' needs. It aims at defining a compromise between flexibility and complexity for the design of high quality timetables. For this purpose, we consider period-dependent travel times and headways, while keeping fixed stopping times in stations.

We also base our model on realistically available data such as planner-defined importance level of transfers and period-dependent headways. Indeed, while origin-destination data is often taken as input for the models in the literature, the complexity induced by path-assignment is such that the option of re-routing passengers during the process of optimization is usually abandoned and the demand considered fixed and inelastic. Therefore, we directly assign a weight to each transfer based on the transit operator's knowledge and experience.

Also, we consider that users are not captive and will not use a transfer requiring more than a certain waiting time limit. This implies that we need to take into consideration both number and quality (waiting time with respect to provided minimum, ideal and maximum waiting times) of transfers.

Our model supports complex line timetable schemes and uses real-world timetables, in which itineraries can vary with the line runs. This includes skipping or adding stops to the "main" line itinerary in some runs, serving stops in variable order, and lines with branches (when several patterns of itinerary are used, serving different stops, usually at extremities of the line).

2.2 Input

The problem considered in this paper has the following inputs.

- The lines network structure, composed of ordered sets of stops with a planner-defined main stop for each line.
- The current timetable (it is assumed the lines are already in use), comprising for each line run the set of arrival and departure times of the served stops.
- For each line, a set of headway periods with associated variation margins.
- The constitution of groups of lines on which interlining is allowed. Each vehicle will be assigned to line runs exclusively inside the same lines group. This allows to cover the cases in which portions of various sizes of the network are serviced by the same operator, as well as the common urban case of vehicles running back and forth on a single line.
- All needed information concerning external lines are also available, namely connecting point and times with the network.
- The set of parameterized transfers. Parameters include relative level of importance, and minimum, ideal and maximum waiting time for users.
- The deadhead running times between all line run termini of each lines group, which are used in the vehicle assignment part of the problem.

2.3 Variables

The set of *decision variables* is composed of all the line runs on the timetable. The value to be assigned to each variable is a (starting-time, vehicle) pair. The domain of these variables is discrete and finite. The planning horizon is comprised in a time-frame usually of one day long, and discretised in minutes without loss of generality. The number of vehicles is at most equal to the number of line runs in the timetable.

Since we use fixed stopping times and the period-dependent running times are also given, we can then deduce all arrival and departure times for the rest of the stops from the starting time of each line run.

2.4 Constraints

Feasibility Constraints. A feasible solution must meet three conditions:

- The stopping time at each stop is equal to the initial stopping time at the same stop of the same line run.
- The running times between stops match the period-dependent data.
- In any vehicle schedule, the time gap separating the arrival at the last stop of a line run and the departure from the first stop of the next line run assigned to it must be greater or equal to the turnaround time of the terminal plus the deadheading trip duration.

Timetable Structure. Within a given line, the set of stops served and their order can vary with the runs along the day. The structure of each run is fixed, and the order in which the runs are served cannot be modified. Additionally, the first (resp. last) run of a line cannot serve the main stop before (resp. after) the start (resp. end) time of the first (resp. last) headway period of the line.

Complete Assignments. A (starting-time, vehicle) couple value must be assigned to each decision variable. In order to be consistent, we require the assignment to be complete such that:

- A departure time must be assigned to each line run.
- One vehicle must be assigned to each line run.

Group Interlining. All resources must be assigned to line runs belonging to the same line group.

2.5 Objectives

Fleet Size. The number of vehicles is the main resource objective.

Number and Quality of Transfer Possibilities. As mentioned in 2.1, we base our transfer quality evaluation on bounds and ideal value provided by the planner. The cost function we use is a nonlinear function of the waiting time which favors the most heavily close-to-ideal waiting times. The cost incurred to the configuration is also pondered by the relative level of importance of the transfer. The time gaps between arriving and departing runs belonging to lines meeting in the network are computed. Each gap belonging to the allowed interval means a transfer opportunity and generates a negative cost to the configuration.

Headway Evenness. In the context of a minimization problem, we model this objective as one of minimizing the sum of evenness defaults. Dealing with multiple headway periods and their transitions is a challenging problem [2]. Let us describe the basics of our method to deal with the three types of situations which can arise (see Fig.1):

- *Case 1: Consecutive runs belong to the same headway period.* The observed interval should be as close as possible to the expected headway.
- *Case 2: Consecutive runs belong to adjacent headway periods.* Only gaps shorter than both or longer than one of the expected headways are penalized.
- *Case 3: Observed interval with start and end of the day.* For each line, two of these intervals occur: between the allowed start of the day and the first actual run, and between the last run and the end of the day. It is assumed that these values can be "too long" but not "too short", and we compare them to the variation margin.

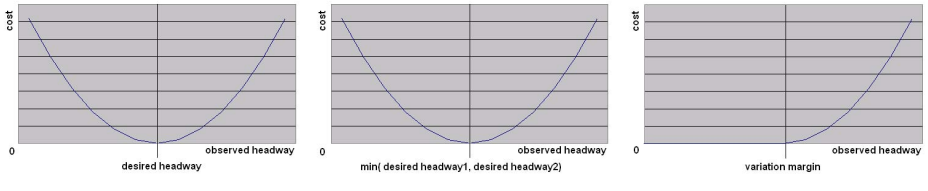


Fig. 1. Individual headway cost functions incurred respectively for case 1, 2 and 3

3 Solution Approach

Our transit network timetabling and vehicle scheduling problem enables several important features: complex timetable schemes, multiple headway periods for each line and variable running times along the day. These features induce at the same time additional difficulties for solving the problem. Given the intrinsic complexity of the model, we choose to employ a heuristic solution approach rather than exact methods. For this purpose, we use Iterated Local Search for its simplicity and efficiency.

In what follows, we recall the basic idea of ILS and the algorithm used for the vehicle assignment, then introduce the evaluation function and the neighborhoods employed, and from there we present the developed approach.

3.1 Iterated Local Search

Iterated Local Search (ILS) is a simple and robust metaheuristic [10]. It is based on the principles of Local Search combined with perturbation movements that are used when the search is caught in a local optimum. If this perturbation satisfies a given acceptance criterion, another round of local search is applied to the current solution, eventually leading to another local optimum. These alternate phases of intensification and diversification permit an exploration of the local optima of the search space that can provide effective results. In our case, we use "Markovian" walk dynamics rather than a history. This choice is based on the fact that the defined perturbation movements diversifies the search enough to limit the probability of leading back to the last local optimum.

3.2 Vehicle Assignment

The vehicle assignment part of our problem, i.e. the linkage of runs, can be modeled as a network-flow-based *quasi-linear assignment problem* and solved optimally by an efficient auction algorithm [8]. This algorithm consists in assigning the source and the trips to trips or to the sink (in order to create sequences). We assign half the cost of a vehicle to the links between the source and trips and between trips and the sink. The cost is somewhat different from Freling's model of [8] in that we do not include in this cost any value related to the deadheading time for pull-in and pull-out trips, since information regarding the depots is unavailable. Therefore all the links leaving the source and entering the sink have the same value. The other "intermediate" links, joining two trips, are assigned a value depending on the feasibility of the connection. The algorithm gives optimal results in a very short time, especially on sparse networks. This fits well our context of limited groups of lines for the vehicle assignment.

3.3 Evaluation Function

An aggregated weighted sum is used to evaluate the global cost of a solution. While the number of vehicles involved and number of feasible transfers are

countable, notions of transfer quality and headway evenness are more fuzzy. Non-linear cost functions are defined for these objectives (see 2.5). This evaluation function is used in the context of our minimization problem.

3.4 Moves and Neighborhoods

RunShift: Recall that the value of a variable is composed of a (starting-time, vehicle) couple. The $RunShift$ move modifies only the time value of one single variable and the vehicle values of none to all of the variables.

Given a solution (i.e. a timetable and a schedule), a neighboring solution can be obtained by shifting the departure time of a single randomly selected line run by +/- n minutes. n is chosen randomly inside a restricted interval defined to both respect the timetable structure constraint and prevent large shifts that are likely to incur high costs on the headway evenness objective.

The vehicle assignment is then recomputed on the neighboring timetable, resulting in the possible modification of the vehicle value of up to the entire set of variables. These values are determined in order for the schedule to remain inside the realisability domain by respecting the sequence feasibility and group interlining constraints, and also to be cost-optimal on the vehicle assignment problem.

LineShift: The $LineShift$ move modifies the time value of a definite set of variables at a time and the vehicle values of none to all of the variables.

A neighboring solution can be obtained by shifting the departure time of all the runs of one single randomly selected line by +/- n minutes. As in *RunShift*, n is chosen randomly inside a restricted interval defined to both respect the timetable structure constraint and prevent excessively large shifts that could be detrimental to the headway evenness objective.

The vehicle assignment is then completely recomputed on the neighbor timetable.

3.5 Initial Solution and General Procedure

Our ILS algorithm is based on a heuristic initial construction and uses two neighborhoods aiming at alternatively intensifying and diversifying the search.

The initial solution is built according to the following three steps.

- First, a departure time is assigned to each line run of the network based on the existing timetable.
- Second, a vehicle is assigned to each line run through the linear assignment algorithm (see section 3.2).
- Third, a descent method combined with the $LineShift$ neighborhood is applied to explore quickly parts of a restricted but diverse search space.

It should be clear that this initial solution corresponds to a local optimum with respect to the small-sized $LineShift$ neighborhood. At this point, we use a larger neighborhood ($RunShift$) to intensify the search in the vicinity of this local optimum. Additional areas (that were not reachable with $LineShift$) of

the search space (which is now complete) can be explored through $RunShift$ moves. This is the heart of the ILS: at each iteration, a descent method is applied with $RunShift$ up to a local optimum, at which point a perturbation is applied.

This perturbation consists of a short sequence of $LineShift$ moves applied in the context of a descent method. Moves from $LineShift$ are likely to modify substantially the current solution. Therefore, in order not to loose too many of the good properties acquired so far, only moves with negative or null impact on the evaluation function are accepted. The descent is stopped when a number of moves have been accepted or evaluated, or a time limit has been reached.

The acceptation criterion is met when the quality of the current solution is equal to or better than the quality of the best local optimum recorded so far. The stopping criterion of the whole ILS algorithm relies on computational time, number of iterations, and number of iterations without improvement.

4 Experimentations and Numerical Results

4.1 Data Set

Our experimentations are based on a real extraurban transit network of a large French area involving 3 medium-size cities and numerous villages. The network is composed of 50 oriented lines serving 673 stops. Each line is assigned to one of three different operators, represented in our model by line groups (of 8, 16 and 26 lines). On the typical day of operation considered in this study, 318 line runs are scheduled. Additionally, 30 external activities (train and school flows) interact with the bus network. Considering only the spatial structure of the transit network, 282 different kinds of intramodal and intermodal transfers can hypothetically be generated.

4.2 Computational Results

For this particular test, we carried out 10 runs on the transit network instance, allowing for each run 10 minutes of CPU time (corresponding to a reasonable time cutoff on real situations). Our algorithm was coded in C++, compiled with VC++ 9.0, on a laptop equipped with a 1.73 Ghz Intel(R) Pentium(R) M and 1Gb RAM running Windows XP. Averaged results are plotted in Fig. 2 and 3.

Fig. 3 discloses more details of the result. It is easily observed that the algorithm substantially improves both the quality of service (left) and the level of resources (right). The number of vehicles is evaluated at 91 with the current timetable, and reduced to a average of 67 by the algorithm. The number of feasible transfers on the other hand rises from 180 initially to 260 on average. Considering transfer quality and headway evenness, the global cost function related to quality of service is provided in the left graph of Fig. 3. We use a set of parameters that, while considering both aspects of the problem, slightly favors the resources, as is often desired by the regulating authorities. This is why the number of vehicles drops first, and the quality of service only rises afterwards.

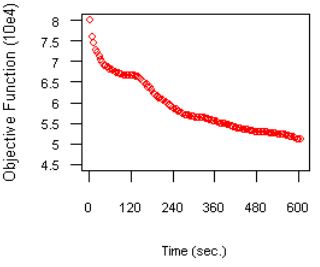
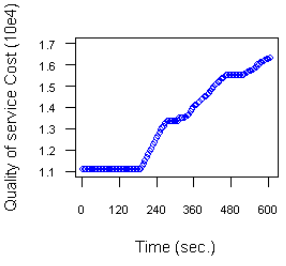


Fig. 2. Evaluation Function Evolution

The first part of the algorithm, based on $LineShift$, stops on average at 130s. We can observe on Fig. 2 that in this short initial period, the descent method based on this neighborhood provides drastic improvement to the solution very fast, and stabilizes at a local optimum. In the second phase consisting of the heart of the ILS, the additional level of detail provided by $RunShift$ permits to explore new areas of the search space and find (important) improvements therein.



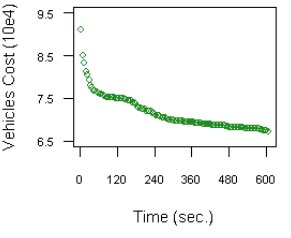


Fig. 3. Profile of the best solution according to the two sides of the problem

5 Conclusion

The problem treated in this paper combines simultaneously two important steps usually treated sequentially in the transit planning process: timetabling and vehicle scheduling. Such a simultaneous approach is, to our knowledge, the first of this kind in the context of this study.

We introduced a natural and high level model in which a departure time and a vehicle must be assigned to each line run of the timetable. This model has the main advantage of being flexible, able to embrace various relevant features of real-world transit networks. In particular, we considered objectives and constraints concerning both quality of service and level of resources as well as other practical features.

A local search optimization procedure is proposed, combining two types of neighborhood in a descent-based Iterated Local Search method. A linear auction algorithm is used to assign vehicles to line runs. Tests carried out on a real and large network showed considerable improvements in both quality of service and level of resources required compared with the current practice. The algorithm has been integrated in a commercial software solution designed for transit operators, and is being used for re-timetabling and scheduling projects by regulating authorities. A path for future work is to integrate features to the solution method that belong to steps both forward and backward in the transit planning process.

Acknowledgments. This work was partially supported by the French Ministry for Research and Education through a CIFRE contract (number 400/2005).

References

1. Bookbinder, J.H., Désilets, A.: Transfer Optimization in a Transit Network. Transportation Science 26, 106–118 (1992)
2. Ceder, A.: Public Transport Timetabling and Vehicle Scheduling. In: Lam, W., Bell, M. (eds.) Advanced Modeling for Transit Operations and Service Planning, pp. 31–57. Elsevier Science Ltd., New York (2003)
3. Cevallos, F., Zhao, F.: Minimizing Transfer Times in a Public Transit Network with a Genetic Algorithm. Transportation Research Record 1971, 74–79 (2006)
4. Chakroborty, P., Deb, K., Sharma, R.K.: Optimal Fleet Size Distribution and Scheduling of Transit Systems using Genetic Algorithms. Transportation Planning and Technology 24(3), 209–226 (2001)
5. Daduna, J.R., Paixao, J.M.P.: Vehicle Scheduling for Public Mass Transit - an Overview. In: Daduna, J.R., Branco, I., Paixao, J.M.P. (eds.) Computer-Aided Transit Scheduling, pp. 76–90. Springer, Berlin (1995)
6. Daduna, J.R., Voss, S.: Practical Experiences in Schedule Synchronization. In: Daduna, J.R., Branco, I., Paixao, J.M.P. (eds.) Computer-Aided Scheduling of Public Transport. LNEMS, vol. 430, pp. 39–55. Springer, Berlin (1995)
7. Deb, K., Chakroborty, P.: Time Scheduling of Transit Systems with Transfer Considerations using Genetic Algorithms. Evolutionary Computation 6(1), 1–24 (1998)
8. Freling, R., Wagelmans, A., Paixao, J.: Models and Algorithms for Single-Depot Vehicle Scheduling. Transportation Science 35(2), 165–180 (2001)
9. Guihaire, V., Hao, J.K.: Transit Network Design and Scheduling: a Global Review. Transportation Research A (to appear)
10. Lourenco, H.R., Martin, O.C., Stutzle, T.: Iterated Local Search. In: Glover, F., Kochenberg, G. (eds.) Handbook of Metaheuristics, pp. 321–353. Kluwer Academic Publishers, Norwell (2002)
11. Quak, C.B.: Bus Line Planning. Master's thesis. Delft University of Techology, The Netherlands (2003)
12. Ting, C.J.: Transfer Coordination in Transit Networks. PhD Thesis, University of Maryland, College Park (1997)

Traveling Salesman Problem and Membership in Pedigree Polytope - A Numerical Illustration

Laleh Haerian Ardekani and Tiru Subramanian Arthanari

Department of ISOM, Business School,
University of Auckland, Auckland 1142, New Zealand
l.haerian@auckland.ac.nz

Abstract. Symmetric traveling salesman problem (STSP), a difficult combinatorial problem is formulated as a multistage insertion (MI) decision problem in Arthanari and Usha (2000). MI formulation is a compact 0-1 formulation for STSP. MI has given rise to the definition of a combinatorial object called pedigree. Arthanari (2008) contains a necessary condition for a MI-relaxation solution to be expressible as a convex combination of pedigrees. The existence of a multicommodity flow with the optimum value equal to unity over some layered network is checked for this purpose. This paper walks through an illustrative example to show the construction of such a network and the procedures involved in checking the necessary condition. Another important feature of this example is it brings out the need for discarding some arcs from the network called dummy arcs, for the correctness of the necessary condition for membership.

Keywords: Traveling salesman problem, Combinatorial optimization, Pedigree polytope, Multistage insertion formulation, Membership problem.

1 Introduction

The Traveling Salesman Problem (TSP) is probably the most studied NP-hard problem in the area of combinatorial optimization [16] and it has served as a very good test bed for validating new algorithms and combinatorial methods [22], [15] and [2].

TSP seeks to find the shortest Hamiltonian tour over a complete graph of a finite set of nodes where the distance between every two nodes is known. In general, TSP intends to *minimize* $C^T x$, subject to $x \in S$, where vector x is indicating to a Hamiltonian tour, with S denoting the set of the incidence vectors of all possible feasible tours and $C^T x$ being the length of the tour x [2],[16].

The standard formulation for TSP was suggested by Dantzig, Fulkerson and Johnson in 1954 [12]. Dantzig, Fulkerson and Johnson formulation (DFJ) is a $0-1$ linear model with each variable representing the existence of an edge in the tour. The number of variables in DFJ is of the order of $O(n^2)$ and the

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 145–154, 2008.

number of the constraints is of the order of $O(2^{n-1})$ which makes it impractical to solve directly, hence Dantzig, Fulkerson and Johnson took resort to cutting off fractional solutions from the solution space by adding violated constraints and re-solving the problem. This method for solving TSP has led to wide spread applications in the combinatorial optimization field. For instance Concorde© software [11] is one of the best known computer codes for TSP that uses branch and cut approach and is capable of solving TSP instances as large as 15000−city problems [2].

There are many different formulations for TSP [12] [13], [10] and [17]. The number of constraints, variables and also solvability of these formulations are different. Padberg and Sung [20] have used a transformation technique to map different formulations from different spaces into the DFJ problem space and then compared the strength of the formulations. Orman and Williams[19] showed that the LP relaxation of DFJ formulation has the smallest polytope and thus is the tightest.

The multistage insertion formulation (MI) suggested by Arthanari [9] has $O(n^2)$ constraints and $O(n^3)$ variables. MI is based on $n-3$ sequential stages of node insertions, from 4 to n, into the subtour $T_3 = [1, 2, 3, 1]$ for constructing a complete $n-$tour . The MI formulation models the TSP by taking these insertions as the decision variables. Let $V_n = \{1, \ldots, n\}$, with n being the number of nodes. Let $E_n = \{e = (i, j) \mid i, j \in V_n, i < j\}$, $\delta(S) = \{(u, v) \in E_n, (u \in S \wedge v \notin S) \vee (u \notin S \wedge v \in S)\}$ for $S \subset V_n$ and also $E(S) = \{(u, v) \in E_n \mid u, v \in S\}$. Let $x_k(e)$ be a 0-1 variable indicating the insertion of city k into the edge e and let $C_k(e)$ be the increase in the length of the tour due to such insertion. The complete MI formulation is:

$$Minimize \sum_{k=4}^{n} \sum_{e \in E_{n-1} \setminus E_3} C_k(e) x_k(e)$$

subject to:

$$x_k(E_{k-1}) = 1, k \in V_n \setminus V_3, \tag{1}$$

$$\sum_{k=4}^{n} x_k(e) = 1, \forall\ e \in E_3, \tag{2}$$

$$-\ x_j(\delta(i) \cap E_{j-1}) + \sum_{k=j+1}^{n} x_k(e) \leq 0, e = (i, j) \in E_{n-1} \setminus E_3, \tag{3}$$

$$x_k(e) \geq 0\ integer, \forall\ e \in E_k, \forall\ k \in V_n \setminus V_3.$$

By relaxing the integer constraint from MI and also adding

$$-\ x_n\Big(\delta(i) \cap E_{n-1}\Big) \leq 0, i = 1, \ldots, n-1, \tag{4}$$

the MI-relaxation problem is achieved. The MI-relaxation polytope is denoted by $P_{MI}(n)$. The affine transformation of $P_{MI}(n)$, projecting out $x_k(e)$ variables, is denoted by $u(n)$. Arthanari and Usha [8] compared $u(n)$ with the subtour elimination polytope $SEP(n)$, and proved that $u(n) \subseteq SEP_n$. Thus MI formulation is as tight as DFJ formulation and has only polynomially many constraints.

Next we consider the definition of pedigree as given in [6]. A pedigree is an integer solution to MI formulation and is an alternative representation of a TSP tour which can be associated with the sequential insertion of nodes into the tours. Pedigree is a combinatorial entity composed of the ordered set of the edges chosen for insertion at each stage throughout the multistage insertion process [6]. A pedigree W corresponding to a tour of n cities is denoted by $W = (e_4, \ldots, e_n)$, where e_k is the edge used for the insertion of city k .

Example 1. For a TSP of the size $n = 7$, the tour $T_7 = [1, 4, 5, 3, 6, 2, 7, 1]$ is equivalent to having the decision variables $x_4((1, 3)) = x_5((3, 4)) = x_6((2, 3)) = x_7((1, 2)) = 1$. The corresponding pedigree is $W = ((1, 3), (3, 4), (2, 3), (1, 2))$. In this pedigree, node 4 was inserted into the edge $(1, 3)$, resulting in the edges $(3, 4)$ and $(1, 4)$. The edge $(3, 4)$ was then used for the insertion of node 5 which resulted in $(3, 5)$ and $(4, 5)$ and so on.

1.1 The Pedigree Polytope

The MI formulation has given rise to the definition of pedigree polytope. The set of all the pedigrees for a TSP of size n, is denoted by P_n and $conv(P_n)$ is the convex hull that is formed by all the pedigrees as its extreme points and is called the pedigree polytope. Each vertex of $conv(P_n)$ corresponds to a unique pedigree and hence a unique TSP tour. $conv(P_n)$ lies on the hyperplanes $x_k(E_{k-1}) = 1, \forall k \ \in V_n \setminus V_3$. The dimension of $conv(P_n)$ is shown to be equal to $\sum_{(k=4)}^{n}(k-2)(k-1)/2 - (n-3)$ in [4].

Pedigree polytope has some interesting characteristics, for instance, the testing of non-adjacency of two pedigrees can be conducted in polynomial time [5] whereas the non-adjacency testing for two tours is shown to be NP-complete [21]. Other characteristics of pedigree polytope are discussed in [5] and [6].

It is shown that the convex hull of pedigrees is the subset of the MI polytope ($conv(P_n) \subset P_{MI}(n)$) [8] therefore it would be interesting to check whether given a feasible solution $X \in P_{MI}(n)$, it also belongs to the pedigree polytope. This is equivalent to checking if a solution from the LP relaxation of MI corresponds to a convex combination of feasible tours. This problem is called the membership problem. The complexity of an optimization problem over a polytope is similar to that of the membership problem of the polytope [14], thus we are interested in solving the membership problem.

It is shown by Arthanari that a necessary condition for a MI-relaxation solution to be expressible as a convex combination of pedigrees can be associated with the existence of a multicommodity flow with the optimum value equal to unity over some layered network [5]. In this construction a procedure is used to identify some arcs as dummy arcs which are deleted from the network. In this paper an example is given to show that discarding dummy arcs is essential for the correctness of the necessary condition. This example is also used to illustrate the procedures involved in checking the necessary condition.

Section 2 walks through the illustrative example. The algorithms and results from [3] are used for checking this necessary condition for membership. And finally, the conclusion and future research are presented in Section 3.

2 The Membership Problem

Given X, an MI-relaxation solution, let X/k denote the solution including cities 1 to k. The membership problem checks that given some MI-relaxation solution X, where X/k belongs to the pedigree polytope $conv(P_k)$, whether $X/k+1$ belongs to the pedigree polytope $conv(P_{k+1})$ or not.

Given $4 \leq k \leq n$, let N_k be the layered network which can be constructed in a recursive fashion in $k-3$ stages. Starting from a subnetwork N_4 including only the first two layers, the layers are added one after another in each stage. Each stage consists of adding the next layer to the network N_k and solving some maximum flow problems in the network to define the arc capacities of the arcs between N_k and the new layer. In the bipartite network of the layers $k-3$ and $k-2$, a maximum flow problem denoted by F_k is then solved. In each stage, given $X/k \in conv(P_k)$, it is checked whether $X/k+1 \in conv(P_{k+1})$ or not. It is said that N_k is well-defined, when $X/k+1 \in conv(P_{k+1})$. In case $k=4$, the process for checking the necessary condition includes solving F_4. The infeasibility of F_4 is sufficient for $X/5$, and thus X, not being a member of the respective pedigree polytopes.

Checking the necessary condition for $k=4$ and $k>4$ is illustrated in Section 2.1 and 2.2 respectively.

2.1 Solving F_4 and Forming N_4

Checking the necessary condition for $k=4$ is illustrated through Example 2.

Example 2. Consider X, a MI-relaxation solution corresponding to a 8−city problem, with the following $x_k((i,j))$ values: $x_4((1,2)) = x_4((1,3)) = \frac{2}{5}$, $x_4((2,3)) = \frac{1}{5}$, $x_5((1,2)) = \frac{2}{5}$, $x_5((1,3)) = \frac{3}{5}$, $x_6((1,2)) = x_6((1,5)) = x_6((2,5)) = \frac{1}{5}$, $x_6((3,5)) = \frac{2}{5}$, $x_7((1,4)) = \frac{2}{5}$, $x_7((3,5)) = \frac{1}{5}$, $x_7((5,6)) = \frac{2}{5}$, $x_8((5,6)) = \frac{3}{5}$ and $x_8((6,7)) = \frac{2}{5}$.

The network N_4 includes nodes corresponding to $x_4(e)$ and $x_5(e)$ variables with positive values. Let $[k:i,j]$ be a node in N_4 corresponding to variable $x_k((i,j)) > 0$ and let $V_{[l]}$ be the set of nodes in layer l. In this example $V_{[1]} = \{[4:1,2],[4:1,3],[4:2,3]\}$ and similarly $V_{[2]} = \{[5:1,2],[5:1,3]\}$. Each node $[k:i,j]$ in the network corresponds to the insertion of some city k into the edge (i,j), and therefore each arc between the two layers corresponds to two successive insertions. Let the arc in N_k, connecting a node in layer $k-3$ to a node in layer $k-2$ be denoted as $([k:i,j],[k+1:r,s])$ where $[k:i,j] \in V_{[k-3]}$ and $[k+1:r,s] \in V_{[k-2]}$. In the bipartite network of F_4, nodes of the first layer are treated as origins with supplies equal to $x_k((i,j))$ values and the nodes in the second layer are destinations with demands equal to $x_{k+1}((i,j))$ values.

In the insertion process, since some successive insertions are infeasible, the corresponding arcs in the network are considered to be forbidden and discarded from the network e.g. arc $([4:1,2],[5:1,2])$ corresponds to the insertion of city 4 and 5 into the same edge which is infeasible. Similarly $([4:1,2],[5:3,4])$ is not possible as $(3,4)$ is not available from the insertion of 4 in $(1,2)$. Therefore the set of arcs for F_4 is $\{([4:1,3],[5:1,2]),([4:2,3],[5:1,2]),([4:1,2],[5:1,3]),([4:2,3],[5:1,3])\}$. The capacities of the arcs in F_4 are set equal to one. In this example, F_4 is solved and its optimal value is equal to unity. Let $f([k:i,j],[k+1:r,s])$ denote the flow along the arc $([k:i,j],[k+1:r,s])$. The optimal flow through the arcs of F_4 are $f([4:1,2],[5:1,3]) = f([4:1,3],[5:1,2]) = \frac{2}{5}$ and $f([4:2,3],[5:1,3]) = \frac{1}{5}$.

2.2 Defining Dummy and Rigid Arcs in F_4

If the flow along an arc is same in all feasible solutions of F_k, that arc is called a rigid arc. The rigid arcs along which the flow is zero are called dummy arcs and are discarded from the network. The capacity of the rigid arcs will be set equal to the flow they are carrying. Defining the dummy and rigid arcs can be done by applying a polynomial time algorithm called the frozen flow finding algorithm [3] on the optimal solution of F_4. After applying the frozen flow finding algorithm on N_4, the arcs $([4:1,2],[5:1,3])$, $([4:1,3],[5:1,2])$ and $([4:2,3],[5:1,3])$ are marked as rigid. The arc $([4:2,3],[5:1,2])$ is marked as dummy and therefore discarded from the network. By updating the arc capacities and discarding the dummy arcs, the network we obtain is the network N_4 which is well-defined. And it is concluded that $X/5 \in conv(P_5)$. If F_4 was infeasible we would conclude $X/5 \notin conv(P_5)$ and so $X \notin conv(P_n)$.

Next we define a procedure for checking the necessary condition for $X/k+1 \in conv(P_{k+1})$ given $X/k \in conv(P_k)$ for $k > 4$.

2.3 Checking the Necessary Condition for Membership in the Pedigree Polytope for $k > 4$

The procedure for checking the necessary condition for any $k > 4$ is summarized in the algorithm bellow.

Checking the necessary condition for membership in the pedigree polytope.
Input: MI-relaxation solution X, some $k > 4$, $X/k \in conv(P_k)$ and N_k being well-defined.
Question: Does $X/k+1$ satisfy the necessary condition for membership in $conv(P_{k+1})$?
Output: Yes/No.

Step 1 - Identify links L between layers $k-3$ and $k-2$.
Step 2 - Find capacities for links L solving maximum flow problems in the restricted networks [1] $N_{k-1}(L)$ for each link.

[1] The restricted network for a link is explained through the illustrative example.

Step 3 - Solve F_k.
Step 4 - **If** F_k is not feasible, **goto** Step 9 .
Step 5 - Identify dummy and rigid arcs in F_k.
Step 6 - Construct N_k.
Step 7 - Define and solve the multicommodity flow problem in N_k. **If** the optimal solution is not equal to 1, **goto** Step 9.
Step 8 - The necessary condition for membership is satisfied. **Stop**.
Step 9 - The solution is not a member of the pedigree polytope. **Stop**.

The steps of the algorithm are explained in detail through a numerical illustration for $k = 5$ and X from Example 2.

Example 2 (continued) Checking the necessary condition
$k = 5$

Step 1 **-** The third layer including the nodes in $V_{[3]} = \{[6:1,2],[6:1,5],[6:2,5],[6:3,5]\}$ is added to the network. The links between the second and the third layers are considered. Let $L = ([5:i,j],[6:r,s])$ be a link between the last two layers.

Step 2 **-** Link L between the edges e_1 and e_2 is feasible only if e_1 and e_2 were generated in the earlier stages and were not used for insertion before this stage. Let $N_{k-1}(L)$ be a subnetwork of N_{k-1} regarding link L, including only the nodes that lead to feasible generation of the edges e_1 and e_2 in L. $N_{k-1}(L)$ is also called a restricted network for link L. Figure 1 shows the restricted networks corresponding to links $([5:1,2],[6:1,5])$ and $([5:1,3],[6:1,5])$. A polynomial-time algorithm for defining the nodes of the restricted network is defined in [3].

Let $C(L_i)$ be the optimal solution of the maximum flow problem in $N_{k-1}(L_i)$ networks. $C(L_i)$ values are then used as the capacities of L_i links in the network. The optimal solutions for the maximum flow problems in the restricted networks for $C(L_i) > 0$ are shown in Figure 2 next to the related links.

Step 3 **-** To make sure that no conflicts would be caused between the new capacities when considered all at the same time, F_5 is solved in the bipartite network of the last two layers. The optimal solution of F_5 is equal to one and the optimal flows are also shown in Figure 2 next to the links.

Step 4 **-** Since the optimal solution of F_5 is equal to one, we may proceed to the next step.

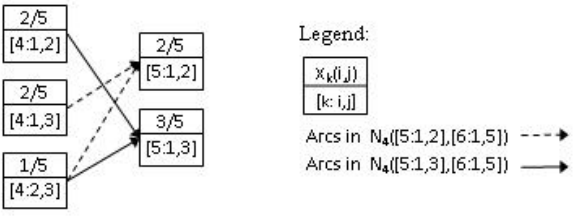


Fig. 1. The restricted networks for two links

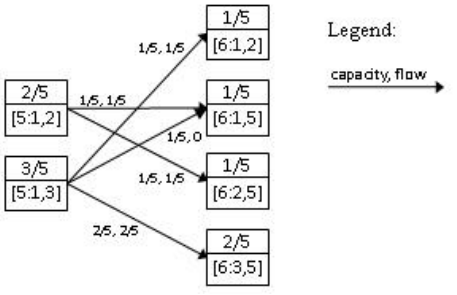


Fig. 2. The capacity and optimal flows in F_5

Step 5 **-** Applying the frozen flow finding algorithm again, the arc $([5 : 1, 3], [6 : 1, 5])$ was identified as dummy and therefore discarded from N_5.

Step 6 **-** A temporary layer of sink nodes are added to N_5. Each sink in this layer corresponds to a link L_i with the demand for a certain commodity i equal to $C(L_i)$. A source node in a temporary layer zero is also added to the network with one unit of supply available for each commodity.

Step 7 **-** The multicommodity flow problem [3] can be solved in the extended network including the sinks and the source node. Like any general network flow problem, multicommodity flow problem includes flow conservation and node and arc capacity constraints. The sink demand constraints in the multicommodity flow problem includes only the rigid arcs in F_k. The multicommodity flow problem in this extended network is solved and the optimal solution is equal to one. Figure 3 shows the flows in the multicommodity flow network.

Step 8 **-** Since the optimal solution to the multicommodity flow problem is equal to one, the necessary condition for membership is satisfied.

It can be observed in Figure 3 that by following the commodity flows, the pedigree paths in the network can be identified. The pedigree paths carrying the commodity flows in this subnetwork are shown in Figure 3. The paths are given in Table 1. Since all the commodity flows in the example are following pedigree paths, and $X/6$ is in fact the convex combination of these pedigrees, the membership in the pedigree polytope $conv(P_6)$ is evident.

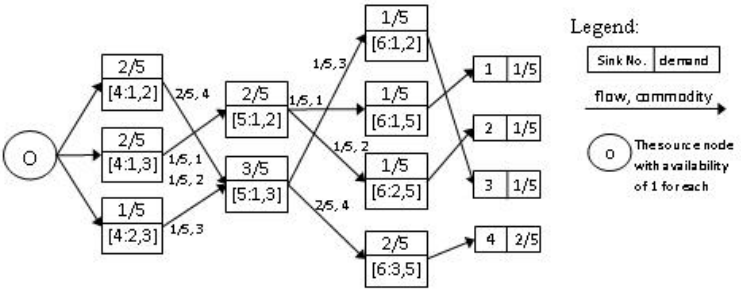


Fig. 3. The N_5 multicommodity flow network

Table 1. The pedigree paths in the N_5 multicommodity flow network

Commodity	Pedigree Path			Flow
	4	5	6	
1	1,3	1,2	1,5	1/5
2	1,3	1,2	2,5	1/5
3	2,3	1,3	1,2	1/5
4	1,2	1,3	3,5	2/5
Total Flow				1

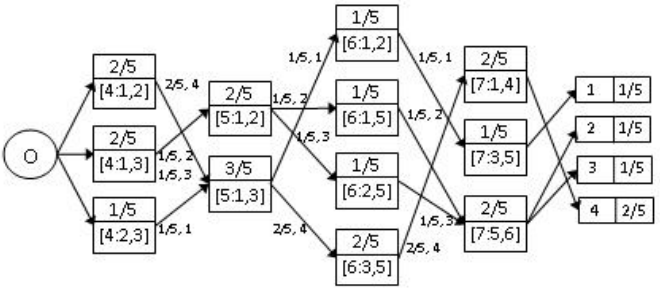


Fig. 4. The well-defined N_6 network

Continuing similarly with $k = 6$ we can check $X/7 \in conv(P_7)$. The corresponding well-defined network N_6 is shown in Figure 4. Finally we trace through the algorithm for $k = 7$.

Step 1 - For $k = 7$, the fifth layer including the nodes in $V_{[5]} = \{[8:5,6],[8:6,7]\}$ is added to the network and the new links between the fourth and fifth layers are considered.

Step 2 - The maximum flow problems in the restricted networks for the links are solved and the optimal solutions are $C(([7:1,4],[8:5,6])) = \frac{2}{5}, C(([7:3,5],[8:5,6])) = \frac{1}{5}$ and $C(([7:5,6],[8:6,7])) = \frac{2}{5}$.

Step 3 - F_7 is solved and the optimal solution was equal to one. The optimal flows are equal to arc capacities.

Step 4 - We may proceed to the next step as the optimal solution of F_7 is equal to one .

Step 5 - All the three links in F_7 are identified as rigid arcs.

Step 6 - Three sink nodes are added to N_7 for each link. The multicommodity flow network for N_7 is formed.

Step 7 - The multicommodity flow problem is solved and the optimal solution is equal to 0.8. The optimal solution is shown in Figure 5.

Step 9 - The solution is not a member of the pedigree polytope.

It should be mentioned that identifying the rigid and dummy arcs is necessary for applying this algorithm. For instance in this example, not discarding the dummy arc ((5,1,3),(6,1,5)) from the network would have resulted in a false optimal solution of one and it might lead to the wrong conclusion that the necessary condition for membership is satisfied.

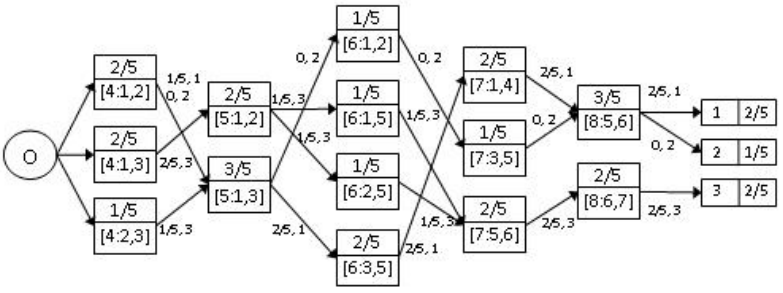


Fig. 5. The N_7 multicommodity flow network

3 Conclusion

It was shown by Arthanari [3] that a necessary condition for membership of the solutions of MI-relaxation in the pedigree polytope can be associated with the existence of a multicommodity flow in some layered network with optimal value equal to unity. Such a layered network is built recursively based on the solution of MI-relaxation.

This paper aims to bring out a work-in-progress on the algorithm for checking the necessary condition for membership. Given a solution from a MI-relaxation instance, the algorithm is traced through and the construction of the layered network is illustrated. A multicommodity flow problem and some maximum flow problems are solved in the network.

Current studies by the authors are targeting MI-relaxation instances of larger sizes. Further computational experiments on bigger problems for the search of possible counter example that shows the necessary condition is not sufficient is currently under progress. Future research is on developing heuristics for proving membership in pedigree polytope using the necessary condition illustrated in this paper and some other sufficient conditions proved in the related papers [5] and [3].

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows Theory: Algorithms and Applications. Prentice Hall, Englewood Cliffs (1993)
2. Applegate, D., Bixby, R.E., Chvatal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton (2006)
3. Arthanari, T.S.: On the Membership Problem of Pedigree Polytope. In: Neogy, S.K., Bapat, R.B., Das, A.K., Parthasarathy, T. (eds.) Mathematical Programming and Game Theory for Decision Making, pp. 61–98. World Scientific, Singapore (2008)
4. Arthanari, T.S.: A Comparison of the Pedigree Polytope with the Symmetric Traveling Salesman Polytope. In: The Fourteenth International Conference of the FIM, Chennai, India, pp. 6–8 (2007)

5. Arthanari, T.S.: On Pedigree Polytopes and Hamiltonian Cycles. Discrete Mathematics 306, 1474–1492 (2006)
6. Arthanari, T.S.: Pedigree Polytope is a Combinatorial Polytope. In: Mohan, S.R., Neogy, S.K. (eds.) Operations Research with Economic and Industrial Applications: Emerging Trends, pp. 1–17. Anamaya Publishers, New Delhi (2005)
7. Arthanari, T.S., Usha, M.: On the Equivalence of the Multistage-Insertion and Cycle Shrink Formulations of the Symmetric Traveling Salesman Problem. Operations Research Letters 29, 129–139 (2001)
8. Arthanari, T.S., Usha, M.: An Alternate Formulation of the Symmetric Traveling Salesman Problem and Its Properties. Discrete Applied Mathematics 98, 173–190 (2000)
9. Arthanari, T.S.: On the Traveling Salesman Problem. In: Bachem, A., et al. (eds.) Mathematical Programming- The State of the Art, p. 638. Springer, New York (1983)
10. Claus, A.: A New Formulation for the Traveling Salesman Problem. SIAM Journal of Algebraic Discrete Methods 5, 21–25 (1984)
11. Concorde Home, http://www.tsp.gatech.edu/concorde/index.htm
12. Dantzig, G., Fulkerson, D., Johnson, S.: Solution of a Large Scale Traveling Salesman Problem. Operations Research 2, 393–410 (1954)
13. Fox, K., Gavish, B., Graves, S.: An n-Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem. Operations Research 28, 1018–1021 (1980)
14. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1988)
15. Junger, M., Reinelt, G., Giovanni, R.: The Traveling Salesman Problem. Network Models. In: Ball, M.O., Magnanti, T.L., Monma, C.L. (eds.) Handbook in Operations Research and Management Science, vol. 7. Elsevier Science, Amsterdam (1995)
16. Lawler, E., Lenstra, J.K., Rinooy Kan, A.H.G., Shmoys, D.B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley, New York (1985)
17. Miller, C., Tucker, A., Zemlin, R.: Integer Programming Formulations and Traveling Salesman Problems. Journal of the Association for Computing Machinery 7, 326–329 (1960)
18. Nemhauser, G., Wolsey, L.: Integer and Combinatorial Optimization. Wiley Interscience, Chichester (1999)
19. Orman, A.J., Williams, H.P.: A Survey of Different Integer Programming Formulations of the Traveling Salesman Problem. In: Kontoghiorghes, E.J., Gatu, C. (eds.) Optimization Econometrics and Financial Analysis. Springer, Heidelberg (2007)
20. Padberg, M., Sung, T.: An Analytical Comparison of Different Formulations of the Traveling Salesman Problem. Mathematical Programming 52, 315–357 (1991)
21. Papadimitriou, C.H.: The Adjacency Relation on the Traveling Salesman Polytope Is NP-Complete. Math. Programming 14, 312–324 (1978)
22. Reinelt, G.: The Traveling Salesman: Computational Solutions for TSP Applications. Springer, Heidelberg (1994)

The Minimum Weight In-Tree Cover Problem

Naoyuki Kamiyama* and Naoki Katoh**

Department of Architecture and Architectural Engineering, Kyoto University,
Kyotodaigaku-Katsura, Nishikyo-ku, Kyoto, 615-8540, Japan
{is.kamiyama,naoki}@archi.kyoto-u.ac.jp

Abstract. Given a directed graph $D = (V, A)$ with a root $s \in V$ such that a non-negative rational weight is associated with each arc in A, we consider the problem for finding a set of minimum weight k spanning in-trees rooted at s which cover A. Here the weight of k spanning in-trees is defined as the sum of weights of all arcs contained in these in-trees. We will show that this problem can be solved in polynomial time. For this, we first consider the set of linear inequalities in $\mathbb{R}^A$ that coincides with the convex hull $P_{\rm ic}(D)$ of a $|A|$-dimensional positive integral vector x such that we can cover A by k spanning in-trees rooted at s such that $e \in A$ is contained in x_e in-trees where $\mathbb{R}$ represents the set of reals. After this, we will show that the separation problem for this polytope can be solved in polynomial time, which implies the polynomial time solvability of the minimum weight in-tree cover problem in conjunction with the ellipsoid method. Furthermore, we will consider the generalization of the minimum in-tree cover problem such that the input directed graph has multiple roots. Although this problem is still open, we give the generalization of the result presented by Vidyasankar [13] which is used to derive the set of linear inequalities which determine $P_{\rm ic}(D)$ to the case of multiple roots.

1 Introduction

Covering problems in a graph are very important from practical and theoretical viewpoints and have been extensively studied. For example, the problem for covering edges in an undirected graph by a minimum number of vertices [9] or a minimum number of cliques [10] is a famous $\mathcal{NP}$-hard problem. On the other hand, the problem for covering all edges of a given undirected graph by minimum number forests can be solved in polynomial time [5,6]. As regards the problems for covering an arc set of a directed graph by subgraphs, for example, the problem for covering arcs in an acyclic directed graph by a minimum number of paths can be solved in polynomial time (see Corollary 14.7a in [12]). The problem for covering all arcs by minimum number branchings can be solved in polynomial time [3]. However, to the best of our knowledge, there do not exist many covering problems which can be solved in polynomial time.

In this paper, we consider the problem for covering a directed graph $D = (V, A, s)$ by k spanning in-trees rooted at s with minimum weight. Here V is a

* Supported by JSPS Research Fellowships for Young Scientists.
** Supported by Grant-in-Aid for Scientific Research (C), JSPS.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 155–164, 2008.

vertex set, A is an arc set, s is a specified vertex $s \in V$ called root and an in-tree is a subgraph T of D such that T has no cycle when the direction of an arc is ignored and all arcs in T are directed to a root. A non-negative rational weight w_e is associated with each arc $e \in A$ and the weight of a spanning in-tree is the sum of weights of arcs in the in-tree and the weight of k spanning in-trees is the sum of weights of k spanning in-trees. Without loss of generality, we assume that s is reachable from every $v \in V$. If the out-degree of s is not zero, it is clear that we can not cover A by k spanning in-trees rooted at s. Thus, we assume that the out-degree of s is zero. Furthermore, since we can always cover A by $|A|$ spanning in-trees rooted at s, we assume that $k \leq |A|$. Then, we consider the problem what we call *minimum weight in-tree cover problem* (in short MWICP) defined as follows.

Problem:	MWICP
Input:	a directed graph $D = (V, A, s)$;
Output:	a minimum weight k spanning in-trees rooted at s which cover A if one exists.

In our recent paper [8], we presented a combinatorial algorithm for finding k spanning in-trees rooted at s which cover A if they exist. The running time of this algorithm is $O(k^7|V|^7|A|^6)$. However, this algorithm does not lead to a polynomial time algorithm for MWICP. To the best of our knowledge, no one studied the problem MWICP, and thus there was no polynomial time algorithm for MWICP.

The problem MWICP can be reformulated as follows. We define the *in-tree cover polytope* $P_{\rm ic}(D) \subseteq \mathbb{R}^A$ by

$$P_{\rm ic}(D) = \text{conv.hull}\left\{ x \in \mathbb{N}^A : \begin{array}{l}\text{We can cover } A \text{ by } k \text{ spanning in-trees rooted}\\ \text{at } s \text{ such that each } e \in A \text{ is contained in } x_e\\ \text{in-trees.}\end{array} \right\}$$

where $\mathbb{R}$ and $\mathbb{N}$ respectively denote the set of reals and positive integers, and for a set X we denote by conv.hull$X$ is the smallest convex set containing X. Then, the problem MWICP is formulated as follows.

$$\min\{\textstyle\sum_{e \in A} w_e x_e : x \in P_{\rm ic}(D)\}. \tag{1}$$

If we can compute an optimal solution x of the problem (1), we can find an optimal solution of the problem MWICP as follows. (i) First, we find k arc-disjoint spanning in-trees $T_1, \ldots, T_k$ rooted at s by using the in-tree packing algorithm of Gabow [4] in the directed graph D' obtained from D by adding $x_e - 1$ parallel arcs to each $e \in A$. (ii) Next, for each arc e' of each T_i, if e is the one created in the previous step (i.e., $e' \notin A$), we replace it by an arc $e \in A$ parallel to e'. The algorithm of Gabow [4] can find k arc-disjoint spanning in-trees rooted at a specified vertex of a directed graph with n vertices and m arcs in $O(k^2n^2 + m)$ time. Since each x_e is clearly at most k, the number of arcs of D' is $O(k|A|)$. Thus, the first step of the above procedure can be done in $O(k^2|V| + k|A|)$ and the time complexity of the second step is clearly

$O(k|V|)$. Hence, in order to prove the polynomial time solvability of MWICP, it is sufficient to show that the problem (1) can be solved in polynomial time.

It is known that the problem (1) can be solved in polynomial time if we can solve the following separation problem what we call SEPARATION for this polytope can be solved in polynomial time (see Theorem 6.36 in [1]).

Problem:	SEPARATION
Input:	a directed graph $D = (V, A, s)$ and a rational vector $x \in \mathbb{R}^A$;
Output:	If $x \in P_{\rm ic}(D)$, "yes". Otherwise, a rational vector $a \in \mathbb{R}^A$ with $\sum_{e\in A} a_e x_e < \sum_{e\in A} a_e y_e$ for every $y \in P_{\rm ic}(D)$.

Our results. We will first present the set of linear inequalities in $\mathbb{R}^A$ that determines $P_{\rm ic}(D)$ which will help to solve SEPARATION. To the best of our knowledge, no one presented the in-tree cover analogue of the *matching polytope*, the *independent set polytope* and so on although the characterizations in terms of inequalities for the existence of k spanning in-trees rooted at s which cover A were presented by Vidyasankar [13] and Frank [3]. After this, we will prove that SEPARATION can be solved in polynomial time. Furthermore, we will consider the generalization of the minimum in-tree cover problem such that the input directed graph has multiple roots. In this paper, we give the generalization of the lemma presented by Vidyasankar [13] which is used to derive the set of linear inequalities which determine $P_{\rm ic}(D)$ to the case of multiple roots. However, unlike the case of a single root, polynomial time solvability is still open. Nevertheless, we believe that our polyhedral characterization will be crucial to develop a polyhedral approach to solve the problem in polynomial time.

Organization. In Section 2, we consider the set of linear inequalities which determine $P_{\rm ic}(D)$ and the separation problem SEPARATION for this polytope. In Section 3, we will consider the generalization of the minimum in-tree cover problem such that the input directed graph has multiple roots.

We conclude this section with necessary definitions and fundamental results.

Directed graphs. Let $D = (V, A, s)$ be a directed graph. For $W \subseteq V$, let $\delta_D(W) = \{e = xy \in A\colon x \in W, y \notin W\}$ where $e = xy$ represents an arc whose tail and head are x and y, respectively. For $v \in V$, we write $\delta_D(v)$ instead of $\delta_D(\{v\})$.

Total dual integrality. Here we introduce the notion of *total dual integrality* presented by Edmonds and Giles [2] which plays a crucial role in this paper. Let A and b an $m \times n$ rational matrix and an m-dimensional rational vector, respectively. For an n-dimensional variable vector x, a system $Ax \geq b$ is called totally dual integral, or just TDI, if for any $c \in \mathbb{Z}^n$ where $\mathbb{Z}$ denotes the set of integers, the dual of minimizing $c^\top x$ over $Ax \geq b$ has an integer optimum solution y, if it is finite. We will use the following lemmas concerning TDI.

Lemma 1 (Corollary 22.1b in [11]). *If $Ax \geq b$ is a TDI system and b is integral, every vertex of the polyhedron $\{x\colon Ax \geq b\}$ is integral.*

Lemma 2 (Theorem 22.2 in [11]). *Let $Ax \geq b$ be TDI and let $A'x \geq b'$ arise from $Ax \geq b$ by adding $-\alpha^\top x \geq -\beta$ for some inequality $\alpha^\top x \geq \beta$ in $Ax \geq b$. Then $A'x \geq b'$ is also TDI.*

Furthermore, a system $Ax \geq b$ is called *box-totally dual integral* or *box-TDI* if for each pair of n-dimensional rational vectors l and u, the system obtained from $Ax \geq b$ by adding $l \leq x \leq u$ is TDI. The following lemma is known.

Lemma 3 (Theorem 22.7 in [11]). *Given a box-TDI $Ax \geq b$, a system obtained from $Ax \geq b$ by adding $l \leq x$ such that l is n-dimensional rational vector is TDI.*

2 An Algorithm for the Problem MWICP

In this section, we prove that the problem (1) can be solved in polynomial time. For a directed graph $D = (V, A, s)$, we first present the set of inequalities which determines $P_{\rm ic}(D)$. Consider the following set of linear inequalities for $x \in \mathbb{R}^A$:

$$\begin{array}{ll} \text{(i)} & x_e \geq 1 \quad \text{for all } e \in A, \\ \text{(ii)} & x(\delta_D(W)) \geq k \text{ for all nonempty } W \subseteq V \setminus \{s\}, \\ \text{(iii)} & x(\delta_D(v)) = k \quad \text{for all } v \in V \setminus \{s\}, \end{array} \tag{2}$$

where for $B \subseteq A$ we define $x(B) = \sum_{e \in B} x_e$.

Lemma 4. *$P_{\rm ic}(D)$ is determined by (2).*

Proof. Let Q be the polytope determined by (2). $P_{\rm ic}(D) \subseteq Q$ immediately follows from the following theorem.

Theorem 1 ([13]). *Given a directed graph $D = (V, A, s)$ and a $|A|$-dimensional positive integral vector $x \in \mathbb{N}^A$, $x \in P_{\rm ic}(D)$ if and only if x satisfies*

$$\begin{array}{l} x(\delta_D(W)) \geq k \text{ for all nonempty } W \subseteq V \setminus \{s\}, \\ x(\delta_D(v)) = k \quad \text{for all } v \in V \setminus \{s\}. \end{array}$$

Next we will show $Q \subseteq P_{\rm ic}(D)$. Since $x \in Q \cap \mathbb{Z}^A$ belongs to $P_{\rm ic}(D)$ from Theorem 1, it is sufficient to prove that every vertex of Q is integral.

Theorem 2 ([3]). *Consider the following set of linear inequalities for $x \in \mathbb{R}^A$:*

$$\begin{array}{l} x_e \geq 0 \quad \text{for all } e \in A, \\ x(\delta_D(W)) \geq k \text{ for all nonempty } W \subseteq V \setminus \{s\}. \end{array} \tag{3}$$

Then, the system (3) is box-TDI.

From Lemma 3 and Theorem 2, the system $\mathcal{S}_1$ obtained by adding $x_e \geq 1$ to the system (3) for every $e \in A$ is TDI. Furthermore, from Lemma 2, the system $\mathcal{S}_2$ obtained by adding $x(\delta_D(v)) \leq k$ to the system $\mathcal{S}_1$ for every $v \in A$ is TDI. Even if we remove $x_e \geq 0$ for every $e \in A$ from $\mathcal{S}_2$, the polytope determined by the resulting system (that is, the system (2)) is the same as that determined by $\mathcal{S}_2$. Hence, it follows from Lemma 1 that every vertex of Q is integral. □

2.1 An Algorithm for the Problem SEPARATION

Next we prove that the problem SEPARATION can be solved in polynomial time.

Lemma 5. *The problem* SEPARATION *can be solved in* $O(|V|\cdot \mathsf{MF}(|V|,|A|))$ *time where* $\mathsf{MF}(n,m)$ *denotes the time required to solve the maximum-flow problem defined on a network with* n *vertices and* m *vertices.*

Proof. Given $x \in \mathbb{R}^A$, in order to test whether $x \in P_{\rm ic}(D)$, we need to check that x satisfies (i), (ii) and (iii) of (2) from Lemma 4. We first prove that if there exists an equality or an inequality violated, we can obtain $a \in \mathbb{R}^A$ such that $\sum_{e\in A} a_e x_e < \sum_{e\in A} a_e y_e$ for every $y \in P_{\rm ic}(D)$ in polynomial time.

Case (i). Assume that there exists $\hat{e} \in A$ such that $x_{\hat{e}} < 1$. Since every $y \in P_{\rm ic}(D)$ satisfies $y_{\hat{e}} \geq 1$, we can obtain a desired $a \in \mathbb{R}^A$ by setting (a) $a_e = 1$ if $e = \hat{e}$, or (b) $a_e = 0$ otherwise.

Case (ii). Assume that there exists a nonempty $\hat{W} \subseteq V \setminus \{s\}$ such that $x(\delta_D(\hat{W})) < k$. Since every $y \in P_{\rm ic}(D)$ satisfies $y(\delta_D(\hat{W})) \geq k$, we can obtain a desired $a \in \mathbb{R}^A$ by setting (a) $a_e = 1$ if $e \in \delta_D(\hat{W})$, or (b) $a_e = 0$ otherwise.

Case (iii). Assume that there exists $\hat{v} \in V$ with $x(\delta_D(\hat{v})) < k$ (resp. $x(\delta_D(\hat{v})) > k$). Since every $y \in P_{\rm ic}(D)$ satisfies $y(\delta_D(\hat{v})) = k$, we can obtain a desired $a \in \mathbb{R}^A$ by setting (a) $a_e = 1$ (resp. $a_e = -1$) if $e \in \delta_D(\hat{v})$, or (b) $a_e = 0$ otherwise.

Next we prove that we can check $x \in P_{\rm ic}(D)$ in a desired time complexity. It is clear that we can check whether x satisfies all conditions of (i) and (iii) in $O(|V| + |A|)$ time. Assuming x satisfies (i) and (iii), in order to check (ii), we consider the network $\mathcal{N}$ which is D with capacity x_e for each $e \in A$. Let $k' = \min\{x(\delta_D(W)) \colon \emptyset \neq W \subseteq V \setminus \{s\}\}$. Then, from the max-flow min-cut theorem [7], it follows that $\min\{\mathsf{MaxValue}(v) \colon v \in V \setminus \{s\}\} = k'$ where $\mathsf{MaxValue}(v)$ represents the maximum-flow value from $v \in V \setminus \{s\}$ to s in $\mathcal{N}$. Hence, by calculating $\mathsf{MaxValue}(v)$ for all $v \in V \setminus \{s\}$, we can find $W_{\min} \subseteq V \setminus \{s\}$ with $x(\delta_D(W_{\min})) = k'$. If $k' \geq k$, the condition (ii) is satisfied. Otherwise, the inequality $x(\delta_D(W_{\min})) \geq k$ is violated. This completes the proof. □

Thus, from Lemma 5, we obtain the following theorem.

Theorem 3. *The problem (1) can be solved in polynomial time.*

We remark that the results in this paper are correct for the case where a weight of an arc is allowed to be negative.

3 Generalization to Multiple Roots

In this section, we will consider the generalization of the minimum in-tree cover problem such that the input directed graph has multiple roots. We call this problem the *minimum weight in-tree cover with multiple roots* (in short MWICP-MR). In this problem, we are given a directed graph $D = (V, A, S, f)$ which consists of a vertex set V, an arc set A, a set of d roots $S = \{s_1, \ldots, s_d\} \subseteq S$ and a function

$f\colon S \to \mathbb{N}$. A function f corresponds to a positive integer k in the single root case. For each $i = 1, \ldots, d$, we define V_i as the set of vertices in V from which s_i is reachable in D, and we define an in-tree rooted at s_i which spans V_i as an *s_i-in-tree*. We define a set $\mathcal{T}$ of subgraphs of D as a *feasible set of in-trees* if $\mathcal{T}$ contains exactly $f(s_i)$ s_i-in-trees for every $i = 1, \ldots, d$. Then, the problem MWICP-MR is defined as follows.

Problem:	MWICP-MR
Input:	a directed graph $D = (V, A, S, f)$;
Output:	a minimum weight feasible set of in-trees covering A if one exists.

Here we introduce necessary definitions for the subsequent discussion. For two distinct vertices $u, v \in D$, we denote by $\lambda(u, v; D)$ the local arc-connectivity from u to v in D, i.e.,

$$\lambda(u, v; D) = \min\{|\delta_D(W)|\colon u \in W, v \notin W, W \subseteq V\}. \tag{4}$$

From Menger's theorem, it is known that $\lambda(u, v; D)$ is equal to the maximum number of arc-disjoint paths from u to v in D (see Corollary 9.1b in [12]). We denote a directed graph obtained by adding an arc set B to A by $D + B$, i.e., $D + B = (V, A \cup B, S, f)$. For $S' \subseteq S$, let $f(S') = \sum_{s_i \in S'} f(s_i)$, and let $f(\emptyset) = 0$. For $v \in V$, we denote by $R_D(v)$ a set of vertices in S which are reachable from v in D. For $W \subseteq V$, let $R_D(W) = \bigcup_{v \in W} R_D(v)$. We define D^* as a directed graph obtained from D by adding a new vertex s^* and connecting s_i to s^* with $f(s_i)$ parallel arcs for every $i = 1, \ldots, d$.

If $|\delta_D(v)| > f(R_D(v)) - f(\{v\} \cap S)$ holds for some $v \in V$, there exists no feasible set of in-trees which covers $\delta_D(v)$ from the definition of a feasible set of in-trees. Thus, we assume in the subsequent discussion that $|\delta_D(v)| \leq f(R_D(v)) - f(\{v\} \cap S)$ holds for every $v \in V$. Since we can always cover by $|A|$ s_i-in-trees the arc set of the subgraph of D induced by V_i, we consider the problem by using at most $|A|$ s_i-in-trees. That is, we assume that $f(s_i) \leq |A|$.

As regards covering a directed graph by a feasible set of in-trees, our recent paper [8] presented a combinatorial algorithm for finding a feasible set of in-trees covering A if one exists whose running time is $O(M^7|A|^6)$ where $M = \sum_{v \in V} f(R_D(v))$. However, this algorithm can not solve MWICP-MR.

In order to prove that MWICP-MR can be solved in polynomial time, it is helpful to produce the set of linear inequalities which determines the *multiple roots in-tree cover polytope* $P_{\mathrm{mric}}(D) \subseteq \mathbb{R}^A$ defined by

$$P_{\mathrm{mric}}(D) = \mathrm{conv.hull}\left\{ x \in \mathbb{N}^A \colon \begin{array}{l} \text{We can cover } A \text{ by a feasible set of in-} \\ \text{trees such that each } e \in A \text{ is contained} \\ \text{in } x_e \text{ in-trees.} \end{array} \right\}.$$

Although the polynomial solvability of MWICP-MR is still open, we present in this section the generalization of Theorem 1 which was used to derive the set of linear inequalities of the polytope for a single root case. This generalizes the result of Vidyasankar [13]. The main result of this section is described as follows.

Theorem 4. *Given a directed graph $D = (V, A, S, f)$ and a $|A|$-dimensional positive integral vector $x \in \mathbb{N}^A$, $x \in P_{\mathrm{mric}}(D)$ if and only if x satisfies*

$$\begin{aligned} &x(\delta_D(W)) \geq f(R_D(W)) - f(W \cap S) \text{ for all } W \subseteq V, \\ &x(\delta_D(v)) = f(R_D(v)) - f(\{v\} \cap S) \quad \text{for all } v \in V. \end{aligned} \tag{5}$$

In order to prove Theorem 4, we will prove some lemmas.

Lemma 6. *Given a directed graph $D = (V, A, S, f)$, $\lambda(v, s^*; D^*) \geq f(R_D(v))$ holds for any $v \in V$ if and only if $|\delta_{D^*}(W)| \geq f(R_D(W))$ holds for any $W \subseteq V$.*

Proof. **If-part.** Assume that $|\delta_{D^*}(W)| \geq f(R_D(W))$ holds for every $W \subseteq V$. Let us fix $v \in V$ and we consider $\lambda(v, s^*; D^*)$. Since $R_D(v) \subseteq R_D(W)$ holds for $W \subseteq V$ with $v \in W$, $f(R_D(W)) \geq f(R_D(v))$ holds for $W \subseteq V$ with $v \in W$. Thus, by this fact and the assumption of the proof, $|\delta_{D^*}(W)| \geq f(R_D(v))$ holds for every $W \subseteq V$ with $v \in W$. Hence, $\lambda(v, s^*; D^*) \geq f(R_D(v))$ holds since $\lambda(v, s^*; D^*) = \min\{|\delta_{D^*}(W)| : v \in W, W \subseteq V\}$ follows from (4).

Only if-part. Assume that $\lambda(v, s^*; D^*) \geq f(R_D(v))$. holds for every $v \in V$. Since $|\delta_{D^*}(W)| \geq f(R_D(W))$ holds for $W = \emptyset$, it is sufficient to consider a nonempty $W \subseteq V$. We define the procedure DominateSequence(W) as follows for each nonempty $W \subseteq V$.

Procedure 1. DominateSequence(W)

```
1: t = 0
2: while W ≠ ∅ do
3:    Set t = t + 1
4:    Choose u ∈ W arbitrarily
5:    Set w_t = u and P_t = {v ∈ W : v is reachable from w_t}
6:    Set W = W \ P_t
7: end while
8: return w_1, ..., w_t and P_1, ..., P_t
```

It is clear that the procedure DominateSequence halts. Let us fix a nonempty $W \subseteq V$, and let $w_1, \ldots, w_t$ and $P_1, \ldots, P_t$ be an output of the procedure DominateSequence(W). For $U, U' \subseteq V$, we define $A[U, U']$ as the set of arcs whose tails and heads are in U and U', respectively. In order to prove the "only if-part", we will show Lemmas 7 and 8.

Lemma 7. *For $j = 1, \ldots, t$, $|A[P_j, V^* \setminus W]| \geq f(R_D(P_j) \setminus R_D(P_1 \cup \cdots \cup P_{j-1}))$.*

Proof. Since every vertex in P_j is reachable from w_j, $R_D(P_j) = R_D(w_j)$ holds. Hence, it is sufficient to prove

$$|A[P_j, V^* \setminus W]| \geq f(R_D(w_j) \setminus R_D(P_1 \cup \cdots \cup P_{j-1})). \tag{6}$$

Since no vertex in $P_{j+1} \cup \cdots \cup P_t$ is reachable from w_j, we have $A[P_j, P_{j+1} \cup \cdots \cup P_t] = \emptyset$. From this equation, a path from w_j to s^* in D^* must use either an arc in

$A[P_j, P_1 \cup \cdots \cup P_{j-1}]$ or an arc in $A[P_j, V^* \setminus W]$. Let N be the maximum number of arc-disjoint paths from w_j to s^* in D^* using arcs in $A[P_j, P_1 \cup \cdots \cup P_{j-1}]$. Since the heads of arcs in $A[P_j, P_1 \cup \cdots \cup P_{j-1}]$ are contained in $P_1 \cup \cdots \cup P_{j-1}$, a path from w_j to s^* in D^* using at least one arc in $A[P_j, P_1 \cup \cdots \cup P_{j-1}]$ must pass through a vertex in $R_D(P_1 \cup \cdots \cup P_{j-1}) \cap R_D(w_j)$. Notice that a path from w_j to s^* in D^* must through a vertex in $R_D(w_j)$. Thus, a path from w_j to s^* in D^* using at least one arc in $A[P_j, P_1 \cup \cdots \cup P_{j-1}]$ must use at least one arc between vertices in $R_D(P_1 \cup \cdots \cup P_{j-1}) \cap R_D(w_j)$ and s^*. Hence, since there exactly exist $f(s_i)$ parallel arcs from each s_i to s^* in D^*, $N \leq f(R_D(P_1 \cup \cdots \cup P_{j-1}) \cap R_D(w_j))$. From this equation and since the maximum number of arc-disjoint paths from w_j to s^* in D^* is at least $f(R_D(w_j))$ by the assumption of the proof, the maximum number of arc-disjoint paths from w_j to s^* in D^* using arcs in $A[P_j, V^* \setminus W]$ is at least

$$f(R_D(w_j)) - f(R_D(P_1 \cup \cdots \cup P_{j-1}) \cap R_D(w_j)) = f(R_D(w_j) \setminus R_D(P_1 \cup \cdots \cup P_{j-1})).$$

From this fact, if (6) does not hold, this contradicts that the maximum number of arc-disjoint paths from w_j to s^* in D^* is at least $f(R_D(w_j))$. This completes the proof. □

Lemma 8. $\sum_{j=1}^{t} f(R_D(P_j) \setminus R_D(P_1 \cup \cdots \cup P_{j-1})) = f(R_D(W))$.

Proof. From $P_1 \cup \cdots \cup P_t = W$, it is sufficient to prove that for every $j' = 1, \ldots, t$

$$\sum_{j=1}^{j'} f(R_D(P_j) \setminus R_D(P_1 \cup \cdots \cup P_{j-1})) = f(R_D(P_1 \cup P_2 \cup \cdots \cup P_{j'})). \quad (7)$$

We prove (7) by induction on j'. For $j' = 1$, (7) clearly holds. Assume that (7) holds for $j' \geq 1$. From the induction hypothesis,

$$\begin{aligned} &\sum_{j=1}^{j'+1} f(R_D(P_j) \setminus R_D(P_1 \cup \cdots \cup P_{j-1})) \\ &= f(R_D(P_1 \cup \cdots \cup P_{j'})) + f(R_D(P_{j'+1}) \setminus R_D(P_1 \cup \cdots \cup R_{j'})) \\ &= f(R_D(P_1 \cup \cdots \cup P_{j'}) \cup R_D(P_{j'+1})) = f(R_D(P_1 \cup \cdots \cup P_{j'} \cup P_{j'+1})). \quad (8) \end{aligned}$$

The last equality holds from $R_D(U) \cup R_D(U') = R_D(U \cup U')$ for $U, U' \subseteq V$. □

From Lemmas 7 and 8, we have

$$\begin{aligned} |\delta_{D^*}(W)| &= \sum_{j=1}^{t} |A[P_j, V^* \setminus W]| \quad (\text{since } P_1, \ldots, P_t \text{ is a partition of } W) \\ &\geq \sum_{j=1}^{t} f(R_D(P_j) \setminus R_D(P_1 \cup \cdots \cup P_{j-1})) \quad (\text{from Lemma 7}) \\ &= f(R_D(W)) \quad (\text{from Lemma 8}). \end{aligned}$$

This proves the "only if-part". □

From Lemma 6, we can obtain the following lemma.

Lemma 9. *Given a $|A|$-dimensional positive integral vector $x \in \mathbb{N}^A$, letting $B = \{x_e - 1$ copies of $e \in A\}$, $\lambda(v, s^*; D^* + B) \geq f(R_D(v))$ holds for every $v \in V$ if and only if $x(\delta_D(W)) \geq f(R_D(W)) - f(W \cap S)$ holds for every $W \subseteq V$.*

Proof. Since every arc in B is parallel to some arc in A, $D^* + B = (D + B)^*$ and $R_D(v) = R_{D+B}(v)$ for every $v \in V$ hold. Hence,

$$\begin{aligned}
&\lambda(v, s^*; D^* + B) \geq f(R_D(v)) \text{ for all } v \in V \\
\Leftrightarrow\ &\lambda(v, s^*; (D + B)^*) \geq f(R_{D+B}(v)) \text{ for all } v \in V \\
\Leftrightarrow\ &|\delta_{(D+B)^*}(W)| \geq f(R_{D+B}(W)) \text{ for all } W \subseteq V \text{ (from Lemma 6)} \\
\Leftrightarrow\ &|\delta_{(D+B)^*}(W)| \geq f(R_D(W)) \text{ for all } W \subseteq V. \qquad (9)
\end{aligned}$$

Furthermore, since B contains $x_e - 1$ copies of each $e \in A$, for every $W \subseteq V$

$$\begin{aligned}
|\delta_{(D+B)^*}(W)| &= |\delta_{D+B}(W)| + f(W \cap S) \\
&= |\delta_D(W)| + \textstyle\sum_{e \in \delta_D(W)}(x_e - 1) + f(W \cap S) \\
&= x(\delta_D(W)) + f(W \cap S). \qquad (10)
\end{aligned}$$

Thus, from (9) and (10), the lemma follows. □

Moreover, our recent paper [8] proved the following lemma.

Lemma 10 ([8]). *Given a directed graph $D = (V, A, S, f)$ and a $|A|$-dimensional positive integral vector $x \in \mathbb{N}^A$, $x \in P_{\mathrm{mric}}(D)$ if and only if $B = \{x_e - 1$ copies of $e \in A\}$ satisfies (a) $|B| = \sum_{v \in V} f(R_D(v)) - (f(S) + |A|)$, and (b) $\lambda(v, s^*; D^* + B) \geq f(R_D(v))$ holds for every $v \in V$.*

Based on the above lemmas, we now give the proof of Theorem 4.

Proof (Theorem 4). **If-part.** Assume that x satisfies (5). From Lemma 10, it is sufficient to prove that $B = \{x_e - 1$ copies of $e \in A\}$ satisfies (a) and (b) in Lemma 10. From the second condition of (5),

$$\begin{aligned}
|B| &= \textstyle\sum_{e \in A}(x_e - 1) = \sum_{v \in V} x(\delta_D(v)) - |A| \\
&= \textstyle\sum_{v \in V} f(R_D(v)) - (\sum_{v \in V} f(\{v\} \cap S) + |A|). \qquad (11)
\end{aligned}$$

Since $\sum_{v \in V} f(\{v\} \cap S) = f(S)$ clearly holds, it follows from (11) that B satisfies (a). Furthermore, from Lemma 9 and the first condition of (5), B satisfies (b).
Only if-part. Assume that $x \in P_{\mathrm{mric}}(D)$. Then, from Lemma 10, $B = \{x_e - 1$ copies of $e \in A\}$ satisfies (a) and (b) in Lemma 10. From Lemma 9 and (b), x satisfies the first condition of (5). From the first condition of (5) for each $v \in V$,

$$\begin{aligned}
|B| &= \textstyle\sum_{e \in A}(x_e - 1) = \sum_{v \in V} x(\delta_D(v)) - |A| \\
&\geq \textstyle\sum_{v \in V} f(R_D(v)) - (\sum_{v \in V} f(\{v\} \cap S) + |A|) \quad \text{(from (5))} \\
&= \textstyle\sum_{v \in V} f(R_D(v)) - (f(S) + |A|). \qquad (12)
\end{aligned}$$

Since (12) holds with equality by (a) of Lemma 10, x satisfies the second condition of (5). □

From Theorem 4, in order to consider the set of linear inequalities which determine $P_{\mathrm{mric}}(D)$, we need to consider the following problem.

Is the following system for $x \in \mathbb{R}^A$ box-TDI?

$$\begin{array}{ll} x_e \geq 0 & \text{for all } e \in A, \\ x(\delta_D(W)) \geq f(R_D(W)) - f(W \cap S) & \text{for all } W \subseteq V. \end{array} \tag{13}$$

If this problem can be positively solved, we can show that the following system (14) determines $P_{\mathrm{mric}}(D)$ in the same manner as the single root case.

$$\begin{array}{ll} x_e \geq 1 & \text{for all } e \in A, \\ x(\delta_D(W)) \geq f(R_D(W)) - f(W \cap S) & \text{for all } W \subseteq V, \\ x(\delta_D(v)) = f(R_D(v)) - f(\{v\} \cap S) & \text{for all } v \in V. \end{array} \tag{14}$$

If $f(R_D(W)) - f(W \cap S)$ is a supermodular function on $W \subseteq V$, this problem can be solved in the same manner as in the proof of Lemma 2. However, since $f(R_D(W))$ and $f(W \cap S)$ are respectively submodular and modular functions on $W \subseteq V$, $f(R_D(W)) - f(W \cap S)$ is a submodular function on $W \subseteq V$. Hence, we need a different technique.

References

1. Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial Optimization. John Wiley & Sons, Chichester (1997)
2. Edmonds, J., Giles, R.: A min–max relation for submodular functions on graphs. Annals of Discrete Mathematics 1, 185–204 (1977)
3. Frank, A.: Covering branchings. Acta Scientiarum Mathematicarum [Szeged] 41, 77–81 (1979)
4. Gabow, H.N.: A matroid approach to finding edge connectivity and packing arborescences. J. Comput. Syst. Sci. 50(2), 259–273 (1995)
5. Gabow, H.N., Westermann, H.H.: Forests, frames, and games: Algorithms for matroid sums and applications. Algorithmica 7(5&6), 465–497 (1992)
6. Gabow, H.N.: Algorithms for graphic polymatroids and parametric $\bar{s}$-sets. J. Algorithms 26(1), 48–86 (1998)
7. Ford Jr., L.R., Fulkerson, D.R.: Maximum flow through a network. Canadian Journal of Mathematics 8, 399–404 (1956)
8. Kamiyama, N., Katoh, N.: Covering directed graphs by in-trees. In: Hu, X., Wang, J. (eds.) COCOON 2008. LNCS, vol. 5092, pp. 444–457. Springer, Heidelberg (2008)
9. Karp, R.M.: Reducibility among combinatorial problems. In: Thatcher, J.W. (ed.) Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)
10. Orlin, J.: Contentment in graph theory: covering graph with clique. Indigationes Mathematicae 80, 406–424 (1977)
11. Schrijver, A.: Theory of Linear and Integer Programming. J. Wiley & Sons, Chichester (1986)
12. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Springer, Heidelberg (2003)
13. Vidyasankar, K.: Covering the edge set of a directed graph with trees. Discrete Mathematics 24, 79–85 (1978)

On Importance of a Special Sorting in the Maximum-Weight Clique Algorithm Based on Colour Classes

Deniss Kumlander

Department of Informatics, Tallinn University of Technology, Raja St.15, 12617 Tallinn, Estonia
kumlander@gmail.com
http://www.kumlander.eu

Abstract. In this paper a new sorting strategy is proposed to be used in the maximum weight clique finding algorithm, which is known to be the fastest at the moment. It is based on colour classes, i.e. on heuristic colouring that is used to prune efficiently branches by excluding from the calculation formulae vertices of the same colours. That is why the right ordering before colouring is so crucial before executing the heuristic colouring and consequently the main maximum weight clique searching routine. Computational experiments with random graphs were conducted and have shown a sufficient increase of performance considering the type of application dealt with in the article.

1 Introduction

Let $G = (V, E, W)$ be an undirected graph, where V is the set of vertices, E is the set of edges and W is a set of weights for each vertex. A *clique* is a complete subgraph of G, i.e. one whose vertices are pairwise adjacent. The *maximum clique problem* is a problem of finding maximum complete subgraph of G, i.e. a set of vertices from G that are pairwise adjacent. An *independent set* is a set of vertices that are pairwise nonadjacent. A *graph colouring problem* is defined to be an assignment of colour to its vertices so that no pair of adjacent vertices shares identical colours. The *maximum-weight clique problem* asks for a clique of the maximum weight. The *weighted clique number* is the total weight of weighted maximum clique. It can be seen as a generalization of the maximum clique problem by assigning positive, integer weights to the vertices. Actually it can be generalized more by assigning real-number weights, but it is reasonable to restrict to integer values since it doesn't decrease complexity of the problem. This problem is well known to be NP-hard.

The described problem has important economic implications in a variety of applications. In particular, the maximum-weight clique problem has applications in combinatorial auctions, coding theory [1], geometric tiling [2], fault diagnosis [3], pattern recognition [4], molecular biology [5], and scheduling [6]. Additional applications arise in more comprehensive problems that involve graph problems

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 165–174, 2008.

with side constraints. More this problem is surveyed in [7]. In this paper a modification of the best known algorithm for finding the maximum-weight clique is proposed. The paper is organised as follows.

The section 2 describes in details the algorithm to be extended by a new ordering strategy, so readers can understand an essence of the change and the final result. The following section describes the new idea and presents algorithms. The section 4 contains information about conducted tests. The last section concluded the paper and describes open problems.

2 Description of the Algorithm to Be Extended

This section contains a description of an algorithm known to be the best at the moment [12] in finding the maximum weight clique. It is the algorithm that is about to be improved in the paper and therefore it will be described in quite details in order to understand the improvement idea and will be called a base algorithm in the entire text of the paper. The base algorithm is a typical branch and bound algorithm and is a mix of the classical approach proposed by Carraghan and Pardalos in 1990s [8,9] and of a backtracking strategy proposed by Ostergard [10].

2.1 Branch and Bound Routine and Pruning Using Colour Classes

Crucial to the understanding of the branch and bound algorithms is a notation of the *depth* and *pruning formula*. Initially, at the depth 1 we have all vertices of a graph, i.e. $G_1 \equiv G$. Now the algorithm is going to pick up vertices one by one and form a set of vertices that are connected to it forming a new lower depth. This process is normally called *expanding a vertex* and is repeated for each depth. Notice that only vertices existing on the current depth can be promoted to the lower one. Repeating this routine we always will have a set of vertices on the lowest depth that are connected to vertices selected (expanded) on previous depths. Moreover all vertices expanded on different depth are also connected to each other by the expansion logic. That is a way the maximum clique is formed. The more formal illustration will be the following. Suppose we expand initially vertex v_{11}. At the next depth the algorithm considers all vertices adjacent to the vertex expanded on the previous level, i.e. v_{11} and belonging to G_1. Those vertices will form a subgraph G_2. At the depth 3, we consider all vertices (that are at the depth 2, i.e. from G_2) adjacent to the vertex expanded in depth 2 etc. Let v_{d1} be the vertex we are currently expanding at the depth d. That is:

Let's say that G_d is a subgraph of G on a depth d that contains the following vertices: $V_d = (v_{d1}, v_{d2}, \ldots, v_{dm})$. The v_{d1} is the vertex to be expanded. Then a subgraph on the depth d+1 is $G_{d+1} = (V_{d+1}, E)$,
where $V_{d+1} = (v_{(d+1)1}, \ldots, v_{(d+1)k}) : \forall i \;\; v_{(d+1)i} \in V_d$ and $(v_{(d+1)i}, v_{d1}) \in E$.

As soon as a vertex is expanded and a subgraph, which is formed by this expansion, is analysed, this vertex is deleted from the depth and the next vertex of the depth become active, i.e. will be expanded. This should be repeated until

there are vertices that are not analysed and then the algorithm returns to the higher level. The algorithm should stop if all vertices are analysed on the first level.

The branch and bound algorithm by itself is nothing else than an exhaustive search and is very pure from the combinatorial point of view. Therefore it is always accomplished by a special analyses that identifies whether the current depth could produce a bigger clique that the already found one. Such analysis is normally done by so called pruning formula. If $W(d) + Degree(G_d) \leq CBCW$, where $CBCW$ is a size (weight) of the current maximum weight clique, $W(d)$ is a sum of weights of vertices expanded on previous to d depths and $Degree$ is function that defines how much larger the forming clique can become using vertices of the depth (i.e. vertices forming G_d). If this formula holds then the depth is pruned - it is not analysed further and the algorithm immediately returns to the previous level. The main art of different algorithm of this class is setting how the degree function works. The classical approach [8] will just sum up weight of remaining vertices of the depth. The modification made in the base algorithm [12] is applying colour classes. A vertex colouring is found before running the main algorithm and only the highest weight vertex of each colour is included into the degree function calculation during the main algorithm work applying the fact that no more than one vertex of each colour class (independent set) can be included into any clique. Please check the original work for any proves of the previously stated and for more details of the described approach.

2.2 Backtracking and Colour Classes

A backtracking process is widely known in different types of combinatorial algorithm including one proposed by P. Ostergard [10]. The algorithm starts to analyse vertices in the backward order by adding them one by one into analyses on the highest level instead of excluding as others do (although the lower levels work still the same was as the branch and bound one). The main idea of the algorithm is to introduce one more pruning formula - for each vertex starting from the last one and up to the first one a function $c(i)$ is calculated (i is a vertex number), which denotes the weight of the maximum-weight clique in the subgraph induced by the vertices $\{v_i, v_{i+1}, \ldots, v_n\}$. In other words $c(i)$ will be a maximum-weight clique that can be formed using only vertices with indexes are starting from i. So, the original backtracking search [10] algorithm will define that $c(n)$ equals to the weight of v_n and $c(1)$ is the weight of the maximum-weight clique for the entire graph. Obviously the following new pruning formula can be introduced in the backtracking search using the calculated function: if $W(d) + c(i) \leq CBCW$, where $CBCW$ is still a size (weight) of the current maximum weight clique and $W(d)$ is a sum of weights of vertices expanded on previous to d depths.

Colour classes also improve this idea as well it was demonstrated in the base work [12]. The idea is to calculate c function (actually an array) by colour classes

instead of individual vertices. Lets say that the graph colouring before the main algorithm has produced the set of colours $\{C_1, C_2, \ldots, C_n\}$ and vertices are reordered accordingly to their colours. Now, $c(n)$ will equal to the largest weight vertex of $\{C_n\}$, $c(1)$ is still the weight of the maximum-weight clique for the entire graph and $c(i)$ is the weight of the maximum-weight clique in the subgraph induced by the vertices $\{C_i, C_{i+1}, \ldots, C_n\}$. The pruning formula remains the same although the i indicates now the colour class index of the examined vertex instead of the vertex index. Notice that the backtracking order base on the fixed ordering, so vertices colouring and reordering should be done before starting the backtracking order.

3 New Algorithm Including Sorting Strategy

3.1 Sorting

Sorting always played quite a crucial role in many algorithms. Unfortunately the right ordering doesn't guarantee that the final solution could be obtained immediately in problems like finding the maximum-weight clique (at least in nowadays algorithms). The reason is simple - the answer should be proved by revising all other vertices and cases. So even if a solution is obtained during the first search iteration it still takes long to conclude that the already found clique is the maximum one. Despite of this the sorting is still important since could sufficiently affect the performance of an algorithm. Moreover some algorithms use sorting as a core element of their structure in the maximum clique finding routine. The base algorithm only recommends sorting vertices by weights inside each colour class in the decreasing order. This sorting lets just pick up the last vertex per each colour on whatever depth calculating the degree function since ensures that it will always be the maximum weight one among all vertices remaining on that depth in that colour class. This sorting by itself is a sufficient part of the algorithm, but this paper is about to extend this sorting strategy in order to improve the overall efficiency of the algorithm. The complexity produced by introducing into the maximum clique task weights lays first of all in the sufficient variation of weights among vertices. This variation produces situation when one vertex been included into the forming clique gives much more that a set of others. As it was mentioned earlier describing the base algorithm the degree function calculation is conducted basing on the highest weight vertex that appears in each class among remaining in the subgraph on the depth. Therefore a sufficient distribution of high weight vertices among different classes can sufficiently increase the degree calculation result. At the same time, if any algorithm will be able to propose how we could group high weight vertices into same colour classes then we would improve the degree function as one high weight vertex will cover other, similar high weight vertices - once again only the highest weight vertex is used in calculation by the algorithm logic per colour class. Notice that the task formulated earlier is not a pure sorting one, since it should improve the search

basing on colouring. So the heuristic colouring task is the main constraint here. The desired order should appear after the algorithm has:

1. Defined initial sorting
2. Coloured vertices

It is quite common that the colouring is the task that will sufficiently change the order. Therefore we cannot talk here about a precise ordering, but should say "a probabilistic one", i.e. such ordering that will keep the desired ordering after the heuristic colouring is applied with a certain probability. Notice the term heuristic in the previous sentence. Our analysis of the base algorithm source code, which is published in Internet [13], have shown that the initial proposed ordering by weights is dramatically broken by the colouring strategy during which a vertex to be coloured is always moved to the end of the uncoloured vertices line by swapping, so initial positions of the high weights' vertices are lost just after some colouring iterations. This paper proposes that the colouring should be done in such a way that the ordering is kept as long as possible.

The order direction - increasing or decreasing is another interesting topic. Notice that the key technique of the base algorithm is moving backward in the backtrack search. Generally the backtracking algorithm works better if the larger clique is found right in the beginning therefore the paper suggests to order vertices so that the last colour classes (from which the backtracking search will start) will include the higher weight vertices in average.

3.2 Colouring Algorithm for the Maximum Weigh Clique Algorithm

It is well known that the number of colour classes can be sufficiently larger than the size of the maximum clique. That is why most best known algorithms [10,12] are using a greedy colouring as a heuristic one - there is no points to spend time on more precise colouring since even the best colouring will not guarantee to give a number that will be close to the maximum clique size. At the same time the earlier stated wish to keep the initial ordering by weights force us to propose the following algorithm:

Algorithm for the ordering and colouring

Variables:
N - the number of vertices
a - an array with an initial ordering of vertices: a_i contains a vertex number been in the i-th position of that vertices ordering
b - the new ordering after colouring
C_i - a set of vertices coloured by the i-th color

Operations:
!= - a comparison operation called "not equal"
== - a comparison operation "equals"

Step 1. Initial sorting:
Sort vertices by weights in the increasing order producing an ordering array a

Step 2. Initialise:
$i := 0$
$m := N$

Step 3. Pick up a colour:
$i := i + 1$

Step 4. Colour:
For $k := N$ downto 1
If $a_k! = 0$ & there is no such $j : v_j \in C_i, (a_k, v_j) \in E$ then
$a_k := 0, b_m := a_k, m := m - 1, C_i := C_i$ U b_m

if $m == 0$ then go to the "Final sorting" step
Next
Go to step 3

Step 5. Final sorting:
Re-order vertices inside each colour class in the increasing order by weights.

End: Return the new order of vertices b and colouring C.

3.3 Maximum Weigh Clique Algorithm

Algorithm for the maximum - weight clique problem

$CBCW$ - weight of the current best (maximum-weight) clique
d - depth
G_d - subgraph of G formed by vertices existing on depth d and is induced by E
$W(d)$ - weight of vertices in the forming clique
$w(i)$ - weight of vertex i

Step 0. Sorting and colouring (See the above algorithm):
Sort vertices by weights in the increasing order.
Find a vertex colouring starting from the highest weight vertices. Keep the order of uncoloured vertices.
Re-order vertices inside each colour class in the increasing order by weights.

Step 1. Backtrack search runner:
For $n := NumberOfColourClasses$ downto 1
Goto step 2
$c(n) := W$
Next
Go to End

Step 2. Initialization: Form the depth 1 by selecting all vertices belonging to colour classes with an index greater or equal to n.
$d := 1$.

Step 3. Prune: If the current level can contain a larger clique than already found:
If $W(d) + Degree(Gd) \leq CBCW$ then go to step 7.

Step 4. Expand vertex: Select the next vertex to expand on a depth. If all vertices have been expanded or there is no vertices then control if the current clique is the largest one. If yes then save it (including its size as $CBCW$) and go to step 7.
Note: Vertices are examined starting from the first one on the depth.

Step 5. Prune: If the current level can contain a larger clique than already found:
If expanding vertex colour class index $<> n$
If $W(d) + c$(expanding vertex colour class index) $\leq CBCW$ then go to step 7.

Step 6. The next level: Form the new depth by selecting vertices that are connected to the expanding vertex from the current depth among remaining;
$W(d+1) := W(d) + w$(expanding vertex index)
$d := d + 1$;
Go to step 2.

Step 7. Step back:
$d := d - 1$;
if $d == 0$, then return to step 1
Delete the expanded vertex from the analysis on this depth;
Go to step 2.

End: Return the maximum-weight clique.

Note: It is advisable to use a special array to solve the order of vertices to avoid work by changing adjacency matrix during reordering vertices. Besides, instead of removing vertices from a depth, it is advisable to have a cursor that moves from the first vertex on a depth to the last one. All vertices that are in the front of the cursor are in the analyses, while vertices behind the cursor are excluded from it (already analysed).

4 Computational Results and Discussion

It is common to apply tests on two types of case: randomly generated and standard (like for example the DIMACS package for unweighted case of finding

maximum clique problem). Unfortunately there is no such widely adopted standard package for the weighted case although application of maximum clique with weights plays no less important role in industry and health care. Therefore tests to be conducted in this paper will be restricted to randomly generated graphs.

Several algorithms were published since 1975s. The easiest and effective one was presented in an unpublished paper by Carraghan and Pardalos [8]. This algorithm is nothing more that their earlier algorithm [9] for the unweighted case applied to weighted case. They have shown that their algorithm outperforms algorithm their have compared with. Another work, which is quite widely referenced in different sources as the best was published by P. Ostergard [10]. He also has compared his algorithm with earlier published algorithms and has shown his algorithm works better by the publishing time. The last algorithm to be used in the tests in the base one [12] that was described in details earlier. In order to produce comparison results a set of instances where generated and each instance was given to each algorithm and their spent time (on producing a solution) was measured. The table below demonstrates that tests were conducted from densities from 10% to 90% with a step 10%. For each vertices/density case 1000 instances of graphs were generated. Results are presented as ratios of algorithms spent times on finding the maximum clique. Although this presentation is slightly different from common it has one sufficient advantage from our point of view - it gives platform independency, so the same results can be reproduced on any computer and ratios should stay the same. The compared algorithms were programmed using the same programming language and the same programming technique (since all algorithms are quite similar). The greedy algorithm was used to find a vertex-colouring.

PO - time needed to find the maximum-weight clique by Carraghan and Pardalos algorithm [8] divided by time needed to find the maximum-weight clique by P. Ostergard algorithm [10] - an average ratio.

$VColor - BT - w$ - time needed to find the maximum-weight clique by Carraghan and Pardalos algorithm [8] divided by time needed to find the maximum-weight clique by the base [12] algorithm - an average ratio.

New - time needed to find the maximum-weight clique by Carraghan and Pardalos algorithm [8] divided by time needed to find the maximum-weight clique by the new algorithm - an average ratio.

The following table is constructed in such a way to guarantee that each algorithm execution will take at least one second and no more than one hour. That is why the vertices count locates in the second column of the table below - the number of vertices is a dependent parameter (on the density) and is chosen by the time constraint. As the result the number of used vertices the smaller the higher density is. At the same additional tests have shown no sufficient change of results on other number of vertices for each density and it proved from our point results independency from the number of vertices we actually use.

For example, 38.62 in the column marked New means that Carraghan and Pardalos [8] algorithm requires 38.62 times more time to find the maximum-weight clique than the new algorithm proposed in this paper. Presented results

Table 1. Benchmark results on random graphs

Edge density	Vertices	*PO*	*VColor-BT-u*	*New*
0.1	1000	1.01	1.26	1.40
0.2	800	1.25	2.11	2.93
0.3	500	1.58	2.64	3.93
0.4	300	1.71	3.02	4.61
0.5	200	1.78	3.41	5.87
0.6	200	2.07	6.53	10.42
0.7	150	2.37	10.16	18.25
0.8	100	2.98	17.36	38.62
0.9	100	4.51	79.80	293.64

show that the new algorithm performs very well on any density. It is faster than all algorithms we compare with. Especially good results are shown on the dense graphs, where the new algorithm is faster than the Carraghan and Pardalos algorithm [8] in 293 times and than the best known algorithm [12] circa 3 times.

5 Conclusion

In this paper a new fast algorithm for finding the maximum-weight clique is introduced. The algorithm is based on the best know algorithm, which is a branch and bound one, uses a heuristic vertex-colouring in the pruning rules and a backtracking search by colour classes. The algorithm is always better than other best known algorithms that were used in the comparison test. Notice that unlike the unweighted case, the weighted case is much harder to improve the performance and therefore achieved results, like for example one on the dense graphs, where the new algorithm is 3 times faster than the best known algorithm and 300 times faster than the standard benchmarking base one is a remarkable result from our point of view.

References

1. MacWilliams, J., Sloane, N.J.A.: The theory of error correcting codes. North-Holland, Amsterdam (1979)
2. Corradi, K., Szabo, S.: A combinatorial approach for Keller's Conjecture. Periodica Mathematica Hungarica 21, 95–100 (1990)
3. Berman, P., Pelc, A.: Distributed fault diagnosis for multiprocessor systems. In: 20th Annual International Symposium on Fault-Tolerant Computing, Newcastle, UK, pp. 340–346 (1990)
4. Horaud, R., Skordas, T.: Stereo correspondence through feature grouping and maximal cliques. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 1168–1180 (1989)
5. Mitchell, E.M., Artymiuk, P.J., Rice, D.W., Willet, P.: Use of techniques derived from graph theory to compare secondary structure motifs in proteins. Journal of Molecular Biology 212, 151–166 (1989)

6. Jansen, K., Scheffler, P., Woeginger, G.: The disjoint cliques problem. Operations Research 31, 45–66 (1997)
7. Bomze, M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Handbook of Combinatorial Optimization 4. Kluwer Academic Publishers, Boston (1999)
8. Carraghan, R., Pardalos, P.M.: A parallel algorithm for the maximum weight clique problem. Technical report CS-90-40, Dept of Computer Science, Pennsylvania State University (1990)
9. Carraghan, R., Pardalos, P.M.: An exact algorithm for the maximum clique problem. Op. Research Letters 9, 375–382 (1990)
10. Ostergard, P.R.J.: A new algorithm for the maximum-weight clique problem. Nordic Journal of Computing 8, 424–436 (2001)
11. Johnson, D.S., Trick, M.A. (eds.): Cliques, Colouring and Satisfiability: Second DIMACS Implementation Challenge. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26. American Mathematical Society (1996)
12. Kumlander, D.: Some practical algorithms to solve the maximum clique problem. Tallinn University of Technology Press, Tallinn (2005)
13. Kumlander, D.: Network resources for the maximum clique problem, http://www.kumlander.eu/graph

An Extended Comparison of the Best Known Algorithms for Finding the Unweighted Maximum Clique

Deniss Kumlander

Department of Informatics, Tallinn University of Technology, Raja St.15, 12617 Tallinn, Estonia
kumlander@gmail.com
http://www.kumlander.eu

Abstract. This paper conducts an extended comparison of two best known at the moment algorithms for finding the unweighted maximum clique. This test is extremely important from both industry and theoretical perspectives. It will be useful for further developing of those algorithms as clearly demonstrated both algorithms advantages and disadvantages, while industry should consider tests result selecting the best algorithm to be applied in their particular environment.

1 Introduction

Let $G = (V, E)$ be an undirected graph, where V is the set of vertices and E is the set of edges. Two vertices are called to be *adjacent* if they are connected by an edge. A *clique* is a complete subgraph of G, i.e. one whose vertices are pairwise adjacent. An *independent set* is a set of vertices that are pairwise nonadjacent. A *complement graph* is an undirected graph $G' = (V, E')$, where $E' = \{(v_i, v_j)|v_i, v_j \in V, i \neq j, (v_i, v_j) \notin E\}$ - this is a slightly reformulated definition provided by Bomze et al 1999 [2]. A neighbourhood of a vertex v_i is defined as a set of vertices, which are connected to this vertex, i.e. $N(v_i) = \{v_1, \ldots, v_k|\forall j : v_j \in V, i \neq j, (v_i, v_j) \in E\}$. A *maximal clique* is a clique that is not a proper subset of any other clique, in other words this clique doesn't belong to any other clique. The same can be stated about maximal independent set. The *maximum clique problem* is a problem of finding maximum complete subgraph of G, i.e. maximum set of vertices from G that are pairwise adjacent. In other words the maximum clique is the largest maximal clique. It is also said that the maximum clique is a maximal clique that has the maximal cardinality. The *maximum independent set problem* is a problem of finding the maximum set of vertices that are pairways nonadjacent. In other words, none of vertices belonging to this maximum set is connected to any other vertex of this set. A *graph-colouring problem* or a *colouring* of G is defined to be an assignment of colours to the graph's vertices so that no pair of adjacent vertices shares identical colours. So, all vertices that are coloured by the same colour are nothing more than an independent set, although it is not always maximal.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 175–181, 2008.

All those problems are computationally equivalent, in other words, each one of them can be transformed to any other. For example, any clique of a graph G is an independent set for the graph's complement graph G'. So the problem of finding the maximum clique is equivalent to the problem of finding the maximum independent set for a complement graph.

All those problems are NP-hard on general graphs [7] and no polynomial time algorithms are expected to be found. The maximum clique problem has many theoretical and practical applications. In fact, a lot of algorithms contain this problem as a subtask and this is another important applications area for the problem. The first area of applications is data analyses / finding a similar data: the identification and classification of new diseases based on symptom correlation [3], computer vision [1], and biochemistry [11]. Another wide area of applying the maximum clique is the coding theory [5,12]. There are many others areas of the maximum clique application that makes this problem to be important.

2 Introduction into Branch and Bound Algorithms

First of all the branch and bound types of algorithms should be introduced in case the reader doesn't have enough knowledge since understanding of those is a crucial element to understanding main point of algorithms to be described later. Branch and bound type algorithms do analyse vertices one by one expanding those by selecting into the next level of analysis all vertices among remaining that are connected to the expanding one. This next level of expansion is normally named a depth, so initially, at the first depth all vertices of a graph are presented, i.e. $G_1 \equiv G$. Then the algorithm is executed and picks up a vertex one by one, so the first one to analyse will be v_{11}, where the indexes indicate that it is a version from the first depth and is also the first version on that depth. The algorithm will form the depth 2 by listening there all vertices connected to the algorithm considers all vertices adjacent to the v_{11} and belonging to G_1. Those vertices form a subgraph G_2. If G_2 is not empty then the first vertex of that depth will be expanded next - v_21. The depth 3 will contain all vertices from G_2 that are adjacent to v_{21} and by the selection logic those will also are adjacent to the vertex expanded on the first depth, i.e. v_{11}. Let v_{d1} be the vertex to be expanded at the depth d and G_d is a subgraph of G on a depth d that contains the following vertices: $V_d = (v_{d1}, v_{d2}, \ldots, v_{dm})$. Then a subgraph on the depth d+1 is G_d+1 $= (V_{d+1}, E)$, where $V_{d+1} = (v_{(d+1)1}, \ldots, v_{(d+1)k}) : \forall i \;\; v_{(d+1)i} \in V_d$ and $(v_{(d+1)i}, v_{d1}) \in E$.

Continuing this way the algorithm will finally arrive to a depth where no vertices exist. Then the previous depth number is compared to the currently maximum clique size and all vertices expanding at the moment on all previous levels (those are called a forming clique) are saved as the maximum clique if it is larger. Anyway the analysis is returned to the previous level and the next vertex of that level should be expanded now. Let say that we returned to the d-th level and the previous expanded vertex is v_{d1}. Then the next vertex to be expanded will be v_{d2}. This should be continued as long as there are vertices on the depth. The algorithm should stop if all vertices of the first level are analysed.

The branch and bound algorithm by itself is nothing else than an exhaustive search and is very bad from the combinatorial point of view. Therefore it is always used together with a special check allowing to cut branches (cases) that cannot produce any better solution that the current maximum one. This check is called a pruning formula and the classical work [6] in the field of finding the maximum clique suggests using the following:

$$\text{if } d - m + n \leq CBC \tag{1}$$

where d is a depth, m is the current (under analyses) vertex index on a depth, n is the total number of vertices in the depth and CBC is the current maximum clique size. Actually it can be generalised into the following:

$$\text{if } d - 1 + Degree(G_d) \leq CBC \tag{2}$$

where $Degree$ equals to $n + 1 - m$ since d - 1 represents the number of vertices in the forming clique (expanded on previous levels) and $n + 1 - m$ the number of vertices can be potentially included into the clique (and called a degree of the depth/branch).

If this formula holds then the depth is pruned - it is not analysed further and the algorithm immediately returns to the previous depth.

3 Different Levels of Using a Vertex Colouring in Nowadays Algorithms

Here two algorithms to be reviewed that are using vertex colouring for finding the maximum clique on different levels. The first idea is to re-apply a heuristic vertex colouring on each new level of a branch and bound algorithm. Another idea is to apply the colouring only once before the branch and bound routine starts and then use results on the permanent base. There are two representatives of both ways nowadays, which are claimed to be quickest; therefore a comparison of those algorithms is worth to do to identify how different ways affects the performance in different cases to be solved.

3.1 Re-applying a Heuristic Vertex Colouring

This subchapter algorithm is developed by Tomita and Seki [13]. Both this and the next chapter algorithms use the same idea - any colour class is an independent set and therefore no more than one vertex from each colour class can participate in a clique. The pruning formula for the algorithm is still the same:

$$\text{if } d - 1 + Degree(G_d) \leq CBC \tag{3}$$

but $Degree$ here represents the number of existing colour classes (independent sets). The number of existing colour classes is obviously much better estimation than the number of remaining vertices. Therefore the number of analysed subgraphs decreased dramatically. Obviously the exact colouring cannot be used as

it is a task of the same complexity as the maximum clique finding one. Therefore a heuristic colouring is used. The main difference between this algorithm and the next one is an approach to calculating the number of existing colours. This subchapter algorithm finds the heuristic vertex-colouring on each new depth. Besides, it reorders vertices after finding the colouring by colour index in decreasing order. Therefore, instead of calculating the degree each time a new vertex is expanded, the expanding vertex colour index is used as a degree. The pruning formula is reformulated into the following one:

$$\text{if } d - 1 + colour\_index(m) \leq CBC \tag{4}$$

where d is a depth, m is the current (under analyses) vertex index on a depth, and CBC is the current maximum clique size.

3.2 Re-using a Heuristic Vertex Colouring

Here we present an algorithm that obtains a vertex colouring only once and then re-use during its work. The algorithm was developed by Kumlander [10] independently and simultaneously with the previous one. The first step of the algorithm is to obtain a heuristic vertex colouring and re-order vertices by colour classes, so that colour classes will appear one by one in the new vertices order. The algorithm uses the vertex colouring to apply two pruning rules - the direct one and the backtracking one. The backtracking search described below cannot be used for the previous class of algorithm re-colouring on each depth as the backtracking relies on a fixed vertices ordering. Therefore it is a natural part of algorithms from the re-using class. The direct pruning rule is defined using a degree function, which equals to the number of existing colour classes on a depth. The algorithm prunes also:

$$\text{if } d - 1 + Degree \leq CBC \tag{5}$$

where d is a depth, $Degree$ is the depth (subgraph) degree, which is the number of existing colour classes and CBC is the size of the current maximum clique. This algorithm calculates the degree by examining what colour classes exist on a depth. Actually the degree is calculated only ones when the depth is formed and later only adjusted by decreasing on one when the next vertex to be analysed is from another colour class than the previous one. This improves the performance dramatically.

In fact the fixed ordering lets also to apply here one more technique: the backtracking search. It examines the graph vertices in the opposite to the standard branch and bound algorithm's order. The classical vertex level backtracking considers first of all all cliques that could be built using only v_n, then all cliques that could contain v_{n-1} and v_n, and so forth. The general rule - it considers at the i-th step all cliques that could contain$\{v_i, v_{i+1}, v_{i+2}, \ldots, v_n\}$. The core idea here is to keep in memory the size of the maximum clique found for each i-th step (i.e. i-th vertex at the highest level) in a special array b. So $b[i]$ is the maximum

clique for the i-th vertex while searching backward. This allows employing one more pruning formula:

$$\text{if } d - 1 + b[m] \leq CBC \tag{6}$$

Besides the algorithm can stop the backtracking iteration and go to the next one if a new maximum clique is found. Colour classes can improve the backtracking by doing it on the colour classes' level instead of individual vertices. Lets say that vertices are coloured and sorted by colour classes, i.e. $V = \{C_n, C_{n-1}, \ldots, C_1\}$, where C_i is the i-th colour (or we call it the i-th colour class). The algorithm now considers first of all all cliques that could be built using only vertices of the C_1, i.e. of the first colour class, then all cliques that could be built using vertices of C_1 and C_2, and so forth. The general rule - it considers at the i-th step all cliques of $\{C_i, C_{i-1}, \ldots, C_1\}$ vertices. The array b is also used, but the index here is the vertex colour index. So the pruning formula will be:

$$\text{if } d - 1 + b[colour\_index(m)] \leq CBC \tag{7}$$

The stopping condition remains since the maximum clique size of a subgraph formed by $\{C_i, C_{i-1}, \ldots, C_1\}$ is either equal to the maximum clique size of a subgraph formed by $\{C_{i-1}, \ldots, C_1\}$ or is larger on 1.

4 Tests

Here the described algorithms of both classes are analysed on DIMACS graphs, which are a special package of graphs used in the Second DIMACS Implementation Challenge [8,9] to measure performance of algorithms on graphs having different, special structures.

As it has been mentioned earlier, there is a very simple and effective algorithm for the maximum clique problem proposed by Carraghan and Pardalos [6]. This algorithm was used as a benchmark in the Second DIMACS Implementation Challenge [9]. Besides, using of this algorithm as a benchmark is advised in one of the DIMACS annual reports [8]. That's why it will be used in the benchmarking below and is called the "base" algorithm. Results are presented as ratios of algorithms spent times on finding the maximum clique - so the same results can be reproduced on any platforms. Ratios are calculated using the benchmarking algorithm [6]. The larger ratio is the quicker a tested algorithm works as the ratio shows how much quicker the tested one is. The compared algorithms were programmed using the same programming language and the same programming technique. The greedy algorithm was used to find a vertex-colouring.

TS - time needed to find the maximum clique the base algorithm divided by time needed to find the maximum clique by the algorithm re-applying colour classes [13].

VColor-BT-u - time needed to find the maximum clique the base algorithm divided by time needed to find the maximum clique by the algorithm re-using colour classes [10].

* - An original task for this graph is to find the maximum independent set, so the maximum clique is found from the complement graph.

Table 1. Benchmark results on DIMACS graphs

Graph name	Edge density	Vertices	Maximum clique size	*TS*	*VColor-BT-u*
brock200_2	0.50	200	12	2.3	*4.0*
brock200_3	0.61	200	15	*3.3*	3.2
hamming8-4	0.64	256	16	39.9	*7848.3*
johnson16-2-4	0.76	120	8	7.0	*20.9*
keller4	0.65	171	11	6.7	*11.8*
p_hat300-1	0.24	300	8	1.0	*1.3*
p_hat300-2	0.49	300	25	4.8	*6.6*
p_hat500_1	0.25	500	9	0.9	*1.5*
p_hat700_1	0.25	700	11	1.1	*1.9*
sanr400_0.7	0.70	400	21	1.4	*5.6*
2dc.256*	0.47	256	7	6.6	*14.5*

For example, 4.8 in the column marked *TS* means that Tomita and Seki algorithm [13] requires 4.8 times less time to find the maximum clique than the base one. The quickest result of each row is highlighted by the italic font. Presented results show that the *VColor-BT-u* algorithm [10] outperforms the other in most cases. Both reviewed in the paper algorithms are faster than the benchmarking algorithms.

The next test will be conducted on random graphs from densities from 10% to 90% with a step of 10%. 100 instances of graphs have been generated per density and an average ratio is found per algorithm. Here you can see that *VColor-BT-u* looses to Tomita and Seki algorithm practically on all densities.

Table 2. Benchmark results on random graphs

Edge density	Vertices	*TS*	*VColor-BT-u*
0.10	1300	0.8	1.0
0.20	1000	1.4	1.3
0.30	600	1.9	1.5
0.40	500	2.7	1.7
0.50	300	3.3	2.3
0.60	200	5.4	3.5
0.70	150	10.8	5.6
0.80	100	40.9	16.2
0.90	80	200.6	102.1

5 Conclusion

In this paper two currently best known algorithms for finding the unweighted maximum clique are described and what is more important are compared on different graph types. Both algorithms are branch and bound and both are using

colour classes obtained from a heuristic vertex-colouring to find the maximum clique. The main difference is the method of using the colouring. One of those keep re-colouring the graph for each depth formed during the algorithm work and the second does it only once before the core part of the algorithm is executed. The first looses in spending time each time re-colouring and cannot employ backtracking search, while the second looses in precision of colouring the deeper the depth is. Therefore both algorithms have certain disadvantages been both reported as the best known. That is why the comparison test was interested for the industry and theory. Tests were conducted for both DIMACS graphs representing certain important graph types and for randomly generated graphs. The general result is that the re-using colouring algorithm [10] is the better technique is most cases for DIMACS graphs and the re-applying colouring algorithm [13] have shown superb results on random graphs. This let us to conclude that there is no clear winner and tests conducted in the paper should be carefully revisited selecting one or another algorithm to be applied basing on the particular environment it should happen in.

References

1. Ballard, D.H., Brown, M.: Computer Vision. Prentice-Hall, Englewood Cliffs (1982)
2. Bomze, M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Handbook of Combinatorial Optimization, vol. 4. Kluwer Academic Publishers, Boston (1999)
3. Bonner, R.E.: On some clustering techniques. IBM J. of Research and Development 8, 22–32 (1964)
4. Brelaz, D.: New Methods to Color the Vertices of a Graph. Communications of the ACM 22, 251–256 (1979)
5. Brouwer, A.E., Shearer, J.B., Sloane, N.J.A., Smith, W.D.: A new table of constant weight codes. J. IEEE Trans. Information Theory 36, 1334–1380 (1990)
6. Carraghan, R., Pardalos, P.M.: An exact algorithm for the maximum clique problem. Op. Research Letters 9, 375–382 (1990)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, NY (2003)
8. DIMACS, Center for Discrete Mathematics and Theoretical Computer Science. Annual Report (December 1999)
9. Johnson, D.S., Trick, M. (eds.): Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26. American Mathematical Society (1996)
10. Kumlander, D.: Some practical algorithms to solve the maximum clique problem. Tallinn University of Technology Press, Tallinn (2005)
11. Miller, W.: Building multiple alignments from pairwise alignments. Bioinformatics, 169–176 (1992)
12. Sloane, N.J.A.: Unsolved problems in graph theory arising from the study of codes. Graph Theory Notes of New York XVIII, 11–20 (1989)
13. Tomita, E., Seki, T.: An effcient branch-and-bound algorithm for finding a maximum clique. In: Calude, C.S., Dinneen, M.J., Vajnovszki, V. (eds.) DMTCS 2003. LNCS, vol. 2731, pp. 278–289. Springer, Heidelberg (2003)

An Adapted Branch and Bound Algorithm for Approximating Real Root of a Ploynomial

Hoai An Le Thi[1], Mohand Ouanes[2], and Ahmed Zidna[1]

[1] Laboratoire de l'Informatique Théorique et Appliquée,
UFR Scientifique MIM Université Paul Verlaine-Metz,
Ile du Saulcy, 57045 Metz, France
lethi@univ-metz.fr,
zidna@univ-metz.fr
[2] Département de Mathématiques, Faculté des Sciences,
Université de Tizi-Ouzou, Algérie
ouanes_mohand@yahoo.fr

Abstract. In this paper we propose an efficient algorithm based on branch and bound method and reduced interval techniques to approximate real roots of a polynomial. Quadratic bounding functions are proposed which are better than the well known linear underestimator. Experimental result shows its efficiency when facing ill-conditionned polynomials.

Keywords: Global optimization, quadratic upper function, quadratic lower function, root-finding.

1 Introduction

Several fundamental geometrical problems that arise in the processing of curves and surfaces may be reduced computationally by isolating and approximating the distinct real roots of univariate polynomials on finite intervals. Many different approaches for solving a polynomial equation exist [1]. We briefly mention the methods based on deflation techniques [2]. Other ones proceed by subdividing the interval into a sequence of intervals such that each one contains one and only one root of the polynomial [3]. Another interesting study for computing multiple roots of polynomial has been introduced in [4]. In recent years univariate global optimization problems have attracted common attention because they arise in many real-life applications and the obtained results can be easily generalized to multivariate case. Let us mention the works for the Polynomial and Rational functions [9], [16], the Lipschitz functions [11], and those in [7], [8], [15], [17]. Root-finding problem is not an optimization problem, however we can exploit the idea of branch and bound techniques in global optimization for finding roots of a polynomial.

In this paper we propose an adapted branch and bound approach presented in [7] for finding all roots of a polynomial in a power basis. The main idea of our

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 182–189, 2008.

approach consists in constructing quadratic underestimation and/or overestimation functions of the given polynomial f in a successive reduced interval $[a_k, b_k]$ to locate all intervals in which roots of the quadratic bounding functions and roots of f are the same. The algorithm has a finite convergence for obtaining an ε-solution.

The above idea is based on the following reasoning: let Lf_k and Uf_k be a lower and a upper bound of f on $[a_k, b_k]$, then we have :

- if $Lf_k(x) > 0$, then $f(x) > 0 \quad \forall x \in [a_k, b_k]$. This means that the polynomial has no root in this interval ;
- if $Uf_k(x) < 0$, then $f(x) < 0 \quad \forall x \in [a_k, b_k]$ and so the polynomial has no roots in this interval ;
- if $Lf_k(x)$ or $Uf_k(x)$ has one or two roots on the current interval $[a_k, b_k]$ which are not the roots of the polynomial, then these roots are used to reduce the current interval. By the way, when reducing the interval containing all the roots of f , we can locate all sub-intervals that contain the roots of f . These roots are in fact the roots of quadratic underestimating and/or overestimating functions of f on these sub-intervals.

The performance of the proposed procedure depends on the quality of the chosen lower and upper bounds of f. We introduce a quadratic lower bounding function which is better than the well known linear underestimating of f by the theory of approximation [6]. In the same way we introduce a quadratic upper bounding function.

The structure of the rest of the paper is as follows: Section 2 discusses the construction of a lower and an upper bound of a polynomial. Section 3 describes an adapted branch and bound algorithm to approximate the real roots of a polynomial. Section 4 presents some numerical examples for ill-conditionned polynomials while Section 5 contains some conclusions.

2 Quadratic Bounding Functions

We now explain how to construct an upper bound of a function f which is twice continuously differentiable on an interval $[a, b]$.

For $m \geq 2$, let $\{w_1, w_2, ..., w_m\}$ be the pairwise functions defined as in [6]:

$$w_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x_{i-1} \leq s \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{if } x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

We have

$$\sum_{i=1}^{i=m} w_i(x) = 1, \forall x \in [a, b] \quad \text{and} \quad w_i(x_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{otherwise.} \end{cases}$$

Let $L_h f$ be the piecewise linear interpolant to f at the points $x_1, x_2, ..., x_m$:

$$L_h f(x) = \sum_{i=1}^{i=m} f(x_i) w_i(x). \tag{1}$$

The next result from [6] gives a upper bound and a lower bound of f on the interval $[a, b]$, $(h = b - a)$.

Theorem 1. [6] *For all $x \in [a, b]$, we have $|L_h f(x) - f(x)| \leq \frac{1}{8} K h^2$, i.e.,*

$$L_h f(x) - \frac{1}{8} K h^2 \leq f(x) \leq L_h f(x) + \frac{1}{8} K h^2.$$

In [7] the following quadratic lower bounding function of f is proposed:

$$Lf(x) := L_h f(x) - \frac{1}{2} K (x - a)(b - x) \leq f(x), \ \forall x \in [a, b] .$$

It has been proved (see [7]) that this lower bound is better than the affine minorization given in [6]:

$$Lf(x) \geq L_h f(x) - \frac{1}{8} K h^2.$$

In a similar way, we now introduce a concave quadratic upper bounding function of f :

Theorem 2. *For all $x \in [a, b]$, we have*

$$L_h f(x) + \frac{1}{8} K h^2 \geq Uf(x) := L_h f(x) + \frac{1}{2} K (x - a)(b - x) \geq f(x). \tag{2}$$

Proof. Let $E(x)$ be the function defined on $[a, b]$ by

$$\begin{aligned} E(x) &= L_h f(x) + \frac{1}{8} K h^2 - Uf(x) \\ &= \frac{1}{8} K h^2 - \frac{1}{2} K (x - a)(b - x) \\ &= \frac{K}{2} \left[x^2 - (a + b)x + ab + \frac{1}{4} (b - a)^2 \right]. \end{aligned}$$

E is convex on $[a, b]$, and its derivative is equal to zero at $x^* = \frac{1}{2}(a + b)$. Therefore, for any $x \in [a, b]$ we have

$$E(x) \geq \min\{E(x) : x \in [a, b]\} = E(x^*) = 0. \tag{3}$$

Then, the first inequality in (2) holds. Consider now the function ϕ defined on S by

$$\phi(x) := Uf(x) - f(x) = L_h(x) + \frac{1}{2} K (x - a)(b - x) - f(x). \tag{4}$$

It is clear that $\phi''(x) = -K - f''(x) \leq 0$ for all $x \in S$. Hence ϕ is a concave function, and for all $x \in [a, b]$ we have

$$\phi(x) \geq \min\{\phi(x) : x \in [a, b]\} = \phi(a) = \phi(b) = 0. \tag{5}$$

The second inequality in (2) is then proved. □

3 Description of the Adapted Branch and Bound Algorithm

In this section we describe an adapted branch and bound algorithm for approximating the real roots of a polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$ in an interval $[a, b]$. The initial interval which contains all the roots of f can be computed by using the Cauchy or the Knuth method. Let K be a positive number such that $|f''(x)| \leq K$, $\forall x \in [a, b]$. As described above, we construct upper bounds and lower bounds of f on successive reduced intervals $[a_k, b_k]$ of $[a, b]$. More precisely,

- If $f(a_k) > 0$, we construct Lf_k, a convex quadratic underestimating function of f on the interval $[a_k, b_k]$ defined by setting

$$Lf_k(x) = f(a_k)\frac{b_k - x}{h_k} + f(b_k)\frac{x - a_k}{h_k} - \frac{1}{2}K(x - a_k)(b_k - x). \tag{6}$$

  Clearly, if $Lf_k(x)$ has no roots in $[a_k, b_k]$, then $Lf_k(x) > 0$ $\forall x \in [a_k, b_k]$. Consequently, $f(x) > 0$ $\forall x \in [a_k, b_k]$. Hence $f(x)$ has no roots in $[a_k, b_k]$.
- If $f(a_k) < 0$, we construct Uf_k, a concave quadratic overestimating function of f on the interval $[a_k, b_k]$, by setting

$$Uf_k(x) = f(a_k)\frac{b_k - x}{h_k} + f(b_k)\frac{x - a_k}{h_k} + \frac{1}{2}K(x - a_k)(b_k - x). \tag{7}$$

  Similarly as in the above, if $Uf_k(x)$ has no roots in $[a_k, b_k]$, then $Uf_k(x) < 0$ $\forall x \in [a_k, b_k]$. cSo $f(x)$ has no roots in $[a_k, b_k]$.

The recursive algorithm can be given as follows:
Function $S =$ **RootPolynom**(f, n, a, b, ϵ)
Input: f : the polynomial, n : the degree of the polynomial, a, b : the end of the interval $[a, b]$, ϵ : precision of the roots
Output: S - the set of all found roots of f
begin
$S_k = \emptyset$ is an intermediate set
If $(b - a) < \epsilon$ **then** $S = \emptyset$, return S.
Else

1. Compute $f(a)$
2. **If** $f(a) > 0$, **then**
   (a) Construct Lf_k, a quadratic lower bound of f on the interval $[a, b]$
   (b) Solve the equation $Lf_k(x) = 0$
   - **If** $Lf_k(x)$ has no root in $[a, b]$, **then** $S_k = \emptyset$
   - **Else**
     - **If** Lf_k has one root $r_1 \in [a, b]$, **then**
       - **If** $|f(r_1)| < \epsilon$, **then** $S_k = S_k \cup \{r_1\}$;
       - $S_k = S_k \cup RootPolynom(f, n, r_1 + \epsilon, b, \epsilon)$;
     - **If** Lf_k has two roots $r_1 \in [a, b]$ and $r_2 \in [a, b]$, **then**

- **If** $|f(r_1)| < \epsilon$ **then** $S_k = S_k \cup \{r_1\}$;
- **If** $|f(r_2)| < \epsilon$ **then** $S_k = S_k \cup \{r_2\}$;
- $S_k = S_k \cup RootPolynom(f, n, r_1 + \epsilon, r_2 - \epsilon, \epsilon)$

3. **Else** (* $f(a) <= 0$ *)
   (a) Construct Uf_k a quadratic upper bound of f on the interval $[a, b]$;
   (b) Solve the equation $Uf_k(x) = 0$;
   - **If** Uf_k has no root in $[a, b]$, **then** $S_k = \emptyset$
   - **Else**
     If Uf_k has one root $r_1 \in [a, b]$, **then**
     - **If** $|f(r_1)| < \epsilon$, **then** $S_k = S_k \cup \{r_1\}$;
     - $S_k = S_k \cup RootPolynom(f, n, r_1 + \epsilon, b, \epsilon)$;
   - **Else**
     If Uf_k has two roots $r_1 \in [a, b]$ and $r_2 \in [a, b]$, **then**
     - **If** $|f(r_1)| < \epsilon$, **then** $S_k = S_k \cup \{r_1\}$;
     - **If** $|f(r_2)| < \epsilon$, **then** $S_k = S_k \cup \{r_2\}$;
     - $S_k = S_k \cup RootPolynom(f, n, r_1 + \epsilon, r_2 - \epsilon, \epsilon)$;
4. $S = S_k$, return S.

end

3.1 Convergence of the Algorithm

The algorithm terminates if one of the following criteria is satisfied:

1. The length of the current interval $[a_k, b_k]$ is less than ϵ ;
2. The lower or the upper bound of the polynomial has no roots on the current interval $[a_k, b_k]$.

For $\epsilon > 0$, at least one of the two above conditions must be satisfied after a finite number of iterations: if the second condition is violated during the algorithm, then the first condition must be fulfilled after at most $m = \lfloor (b-a)\sqrt{\frac{K}{8\varepsilon}} \rfloor + 1$ iterations (see [7]).

For $\epsilon = 0$, we have the following result.

Theorem 3. *For $h_k = b_k - a_k$, we have*

$$\lim_{h_k \to 0} (Uf_k(x) - f(x)) = 0 \quad \textit{and} \quad \lim_{h_k \to 0} (f(x) - Lf_k(x)) = 0.$$

Proof. As

$$0 \leq Uf_k(x) - f(x) \leq \frac{1}{2} K(s - a_k)(b_k - s) \leq \frac{1}{2} K {h_k}^2,$$

it holds

$$\lim_{h_k \to 0} (Uf_k(x) - f(x)) = 0.$$

In the same way, if we have

$$0 \leq f(x) - Lf_k(x) \leq \frac{1}{2} K(s - a_k)(b_k - s) \leq \frac{1}{2} K {h_k}^2,$$

then

$$\lim_{h_k \to 0} (f(x) - Lf_k(x)) = 0.$$

The proof is complete. □

4 Illustrative Examples and Computational Results

Ill-conditioned dependence of the zeros on the coefficients occurs for many polynomials having no multiple or clustered zeros, the well known example is the polynomial $\prod_{i=0}^{i=n}(x - i/n)$. For a large n, the zeros jump dramatically because of a smaller perturbation of the coefficients [5]. Furthermore, it would not be appropriate to ignore polynomials with multiple zeros like $(x - 1/2)^n$ since they frequently appear in CAGD. We propose to study the behavior of the proposed algorithm with help of these polynomials. Of course, the polynomials are first written in Power basis. The numerical computations were implemented with the $IEEE754$ double precision floating point arithmetic.

1. **Polynomials of the form** $f(x) = \prod_{i=0}^{i=n}(x - i/n)$. The experimental result shows that up to $n = 20$, the proposed algorithm found every root. Beyond $n = 20$, the method start to fail and the results deteriorate. This is due to successive division operations performed by the algorithm in the power basis.
2. **Polynomials of the form** $f(x) = \prod_{i=0}^{i=n}(x - \alpha_i)$ **with** $0 < \alpha_i < 1$**.** The numbers α_i are chosen at random in $[0, 1]$. As in the previous experiences our method found all the zeros with high accuracy (about 10^{-9}). The numbers α_i are arbitrarily chosen. This experience (and the previous) shows that the manner in which the roots are distributed (at random or uniformly) has no influence on the performance of the method. Only the density has an effect on their stability.

$f(x) = \prod_{i=0}^{i=n}(x - i/n)$	$f(x) = \prod_{i=0}^{i=n}(x - \alpha_i)$	$f(x) = x(x - 1/2)^n(x - 1)$
0	0	0
.999999768138546	9.79239766337205E-02	1
5.00005729583799E-02	.999998917211747	0.499999999999
.949999718867712	.197925531155132	
.100000896356013	.949998669127637	
.150001164734556	.89999780620256	
.200002047363825	.247925935504342	
.250002675831615	.29792619161883	
.300003430727458	.849997394440562	
.350004072135139	.347926640047862	
.400004072461467	.397927135990397	
.45000502409906	.799996880187295	
.500005566022001	.447927504621865	
.550006287009764	.497928483243614	
.60000697258511	.749996008511239	
.650007085846317	.547928810935695	
.700007715675635	.597929709584623	
.750008644101933	.699995263507098	
.800008749363372	.647929904566746	
.899998762393359	0.9700110000233	
.849999999999	0.1499999999999	

3. **Polynomials of the form** $f(x) = x(x - 1/2)^n(x - 1)$**.** For these polynomials, the multiplicity n of the value $1/2$ varies from 2 to 13. For any n, the root $1/2$ is found as a simple zero with an excellent error 10^{-16}. For $n = 13$, the results are summarized in the following table :

5 Conclusion

We propose a new method for finding real roots of a polynomial $f(x)$ which is based on the computation of some lower and upper bounds of $f(x)$ and on successive reducing of the initial interval. Facing ill-conditionned polynomial, the experimental results show the efficiency of our algorithm. The roots are found with a good accuracy (relative error magnitude $=10^{-9}$). Note that the multiplicity order can be found by using derivatives of $f(x)$. As the computations are performed in a finite precision arithmetic and rounding errors affect the coefficients of polynomials of high degree, our results deteriorate beyond $n = 20$. But, it has been shown [5] that the Bernstein basis minimizes the condition number which measures the sensibility of the roots through the coefficients perturbation. Our target is to use this base to improve the stability of the proposed algorithm.

References

1. Pan, V.Y.: Solving a polynomial equation: some history and recent progress. SIAM Rev. 39, 187–220 (1997)
2. Jenkis, M.A., Traub, J.F.: A three-stage algorithm for real polynomials using quadratic iteration. SIAM Journal on Numerical Analysis 7, 545–566 (1970)
3. Mourrain, B., Vrahatis, M.N., Yakoubsohn, J.C.: On the complexity of isolating real roots and computing with certainty the topological degree. Journal of Complexity 18, 612–640 (2002)
4. Zeng, Z.: Computing multiple roots of inexact polynomials (manuscript), http://www.neiu.edu/~zzeng/Papers/
5. Farouki, R.T., Rajan, V.T.: Algorithms for polynomials in Bernstein form. Computer Aided Geometric Design 5, 1–26 (1988)
6. De Boor, C.: A practical Guide to Splines Applied Mathematical Sciences. Springer, Heidelberg (1978)
7. Le Thi, H.A., Ouanes, M.: Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint. Rairo-Operations Research 40, 285–302 (2006)
8. Calvin, J., Ilinskas, A.: On the convergence of the P-algorithm for one-dimensional global optimization of smooth functions. Journal of Optimization Theory and Applications 102, 479–495 (1999)
9. Hansen, P., Jaumard, B., Xiong, J.: Decomposition and interval arithmetic applied to minimization of polynomial and rational functions. Journal of Global Optimization 3, 421–437 (1993)
10. Hansen, P., Jaumard, B., Lu, S.H.: Global Optimization of Univariate Functions by Sequential Polynomial Approximation. International Journal of Computer Mathematics 28, 183–193 (1989)

11. Hansen, P., B., Jaumard, B., Lu, S.H.: Global Optimization of Univariate Lipschitz Functions: 2. New Algorithms and Computational Comparison. Mathematical Programming. 55, 273–292 (1992)
12. Moore, R.: Interval Analysis. Prentice-Hall, Englewood Cliffs (1966)
13. Thai, Q.P., Le Thi, H.A., Pham, D.T.: On the global solution of linearly constrained indefinite quadratic minimization problems by decomposition branch and bound method. RAIRO, Recherche Opérationnelle 30, 31–49 (1996)
14. Ratschek, H., Rokne, J.: New Computer Methods for Global Optimization. Wiley, New York (1982)
15. Ratz, D.: A nonsmooth global optimization technique using slopes the one-dimensional case. Journal of Global Optimization 14, 365–393 (1999)
16. Visweswaran, V., Floudas, C.A.: Global Optimization of Problems with Polynomial Functions in One Variable. In: Floudas, A., Pardalos, P.M. (eds.) Recent Advances in Global Optimization, pp. 165–199. Princeton University Press, Princeton (1992)
17. Sergeyev, Y.D.: Global one-dimensional optimization using smooth auxiliary functions. Mathematical Programming 81, 127–146 (1998)

Portfolio Selection under Piecewise Affine Transaction Costs: An Integer Quadratic Formulation

Mohamed Lemrabott[1], Serigne Gueye[1], Adnan Yassine[1], and Yves Rakotondratsimba[2]

[1] Laboratoire de Mathématiques Appliquées du Havre., 25, rue Philippe Lebon, 76600 Le Havre, France
[2] ECE, Ecole Centrale d'électronique., 53, rue de Grenelle, 75007 Paris, France
mouhet12@yahoo.fr,
{serigne.gueye,adnan.yassine}@univ-lehavre.fr,
w_yrakoto@yahoo.com
http://www.springer.com/lncs

Abstract. In this paper we consider the problem of selecting assets for which transaction costs are given by piecewise affine functions. Given practical constraints related to budget and buy-in thresholds, our purpose is to determine the number of each asset i that can produce the maximum return of a portfolio composed of $(n+1)$ assets (one of them is free of risk). The problem is formulated as an integer quadratic problem and afterwards linearized. Some numerical experiments, using Ilog Cplex 10.1, has been performed. They show that the methodology is promising.

Keywords: Piecewise transaction costs; Integer quadratic programming; portfolio selection; Linearization.

1 Introduction

Porfolio selection problems consist, in short, in optimally selecting, within a number of risky and riskless assets, suitable quantities of each asset in such a way to reach a desirable return without exceeding an acceptable risk. The first and most famous portfolio selection model is due to Markowitz (see [11]). In the Markowitz model, the risk (taken as the variance) may be minimized for a fixed level of return, or, symmetrically, the return may be maximized for a fixed level of risk.

In addition to return and risk, many variants of the Markowitz model involved other aspects. Indeed, one can also take into account the transaction costs, the fact that in practice the asset quantities are integer multiple of transaction lots (called "rounds"), and pratical constraints such as budget limit and buy-in threshold. When round lots are considered, Mansini et Speranza [6] propose a Mixed Integer Linear Programming model whose objective function is the mean semi-absolute risk measure of the portfolio. This model has been solved by a linear programming based heuristic. Moreover, continuous transaction costs, proportional to the traded amount, have been considered in Davis and Norman [3].

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 190–196, 2008.

In this paper, we deal with another variant of portfolio selection problem in which $(n+1)$ assets (whose one of them is free of risk) under transaction costs, given by discontinuous piecewise affine functions, have to be optimally selected. Our purpose here is to determine the maximum return and the corresponding number of securities. This problem has been first considered in Lajili-Jarjir and Rakotondratsimba [10]. The authors proposed in this work an optimization mathematical model whose objective function involved the discontinous transaction cost functions. An enumerative scheme has been used to solve the model, and an analytic methodology allows to reduce significantly the portfolio feasible domain. This method is suited when n is small, say $n = 10$, but is impraticable for huge value since the problem of portfolio selection, independently of the risk function, is a NP-Complete problem.

To tackle this problem, we present a global approach in which the portfolio problem with piecewise transaction costs is reformulated as an integer quadratic program and solved with a linearization scheme. Some numerical experiments on asset instances show that the methodology is successful.

2 The Investor Portfolio Selection Model

To present the model we first need the following notations. Let:

- W be the initial wealth available for the investment, so $0 < W$;
- S_i denotes the initial quote at which the security is bought, where $0 < S_i$ and $i \in \{1, .., n\}$;
- ψ_i and ϕ_i be the transaction functions which are applied respectively on buying (at time 0) and selling (at time 1) the securities i where $i \in \{1, .., n\}$;
- V_i be the expected move up of the security i, so $0 < V_i$;
- r be the risk-free interest rate which is applied during the investment period, so $0 \leq r$;
- δ_i, with $0 < \delta_i < 1$, be such that $\delta_i W$ represents the minimal level of cash invested in the risky asset i;
- γ_i, with $0 < \gamma_i < 1$, be such that $\gamma_i W$ represents the maximal level of cash invested in the risky asset i;
- N_i be a nonnegative integer (we write $N \in N^*$) which corresponds to the quantity of security i.

It is important to notice that following the investment in security i, the remaining money C is invested on a saving account with interest rate r.

With the previous notations, the initial wealth of investor is defined as:

$$W = \sum_{i=1}^{n} \Big[N_i S_i + \psi_i(N_i S_i) \Big] + C.$$

And the final wealth at the end of the investment period can be written as follow:

$$\tilde{W} = \sum_{i=1}^{n} \Big[(1+V_i) N_i S_i - \phi_i\big((1+V_i) N_i S_i\big) \Big] + C(1+r).$$

Our aim is to determine the values $\widehat{N}_1, \widehat{N}_2, ..., \widehat{N}_n$, that maximize, under some constraints, the return of the investment. The return is measured by the value: $Ret(N_1, \ldots, N_n) = \frac{\tilde{W} - W}{W} = \frac{1}{W}\sum_{i=1}^{n}\Big[(V_i - r)N_iS_i - \phi_i\big((1+V_i)N_iS_i\big) - (1+r)\psi_i(N_iS_i)\Big] + r$. The optimal quantities $\widehat{N}_i$ may be obtained by solving the following optimization problem:

$$\begin{cases} Max \quad Ret(N_1, \ldots, N_n) \\ \\ s.c. \quad \delta_i W \leq N_iS_i + \psi_i(N_iS_i) \leq \gamma_i W \\ \qquad 0 < \delta_i \leq \gamma_i < 1 : \quad i \in \{1, ..., n\} \\ \qquad \sum_{i=1}^{n} \gamma_i \leq 1 \\ \qquad N_i \in N^* : \quad i \in \{1, ..., n\}. \end{cases} \tag{1}$$

Definition 1
The functions ψ_i and ϕ_i are defined as:

$$\psi_i(x) = \begin{cases} 0 & for \quad x = 0 \\ \alpha_{i0}x + \beta_{i0} & for \quad 0 = a_{i0} < x \leq a_{i1} \\ . & \\ . & \\ . & \\ \alpha_{iM-1}x + \beta_{iM-1} & for \quad a_{iM-1} < x \leq a_{iM} \\ \alpha_{iM}x + \beta_{iM} & for \quad x > a_{iM}. \end{cases}$$

$$\phi_i(x) = \begin{cases} 0 & for \quad x = 0 \\ d_{i0}x + e_{i0} & for \quad 0 = b_{i0} < x \leq b_{i1} \\ . & \\ . & \\ . & \\ d_{iM-1}x + e_{iM-1} & for \quad b_{iM-1} < x \leq b_{iM} \\ d_{iM}x + e_{iM} & for \quad x > b_{iM}. \end{cases}$$

3 An Integer Quadratic Formulation

Since the functions ψ_i and ϕ_i are discontinuous, and the variables N_i are integers, we cannot solve this problem by standard global optimization techniques.

In the scheme that we propose, we first reformulate the discontinuous piecewise affine transaction costs as quadratic functions and apply linearization techniques to solve the resulting integer quadratic programming model. This kind of reformulation has been studied in many references such as in Keha et al [7] on continuous piecewise affine function, or in Gabrel et al [8] dealing with discontinuous step-increasing cost functions of a multicommodity flow problem. Such reformulation is done by adding new binary variables and additional constraints.

Lemma 1. *To reformulate ψ_i, we add $\mu_{i1}, \mu_{i2}, ..., \mu_{iM_i}$ as follows:*

$$\begin{cases} \psi_i(x) = \sum_{t=1}^{M_i} \mu_{it}\big(\alpha_{it}x + \beta_{it} - \alpha_{it-1}x - \beta_{it-1}\big) \\ \qquad = \sum_{t=1}^{M_i} \mu_{it}\big((\alpha_{it} - \alpha_{it-1})x + \beta_{it} - \beta_{it-1}\big) \\ where: \\ \sum_{t=0}^{M_i-1} \mu_{it+1}(a_{it} - a_{it-1}) \le x \le \sum_{t=1}^{M_i} \mu_{it}(a_{it} - a_{it-1}) \\ \mu_{it} \in \{0,1\}: \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\ \mu_{i0} \ge \mu_{i1} \ge ... \ge \mu_{iM_i}: \quad i \in \{1, ..., n\} \\ a_{i(-1)} = a_{i0} = 0: \quad i \in \{1, ..., n\}. \end{cases} \tag{2}$$

Lemma 2. *The same transformation is applicable for ϕ_i. We add the binary variables $\lambda_{i1}, \lambda_{i2}, ..., \lambda_{iM_i}$ as follows:*

$$\begin{cases} \phi_i(x) = \sum_{t=1}^{M_i} \lambda_{it}\big(d_{it}x + e_{it} - d_{it-1}x - e_{it-1}\big) \\ \qquad = \sum_{t=1}^{M_i} \lambda_{it}\big((d_{it} - d_{it-1})x + e_{it} - e_{it-1}\big) \\ where: \\ \sum_{t=0}^{M_i-1} \lambda_{it+1}(b_{it} - b_{it-1}) \le x \le \sum_{t=1}^{M_i} \lambda_{it}(b_{it} - b_{it-1}) \\ \lambda_{it} \in \{0,1\}, \quad t \in \{0, ..., M_i\}: \quad i \in \{1, ..., n\} \\ \lambda_{i0} \ge \lambda_{i1} \ge ... \ge \lambda_{iM_i}: \quad i \in \{1, ..., n\} \\ b_{i(-1)} = b_{i0} = 0: \quad i \in \{1, ..., n\}. \end{cases} \tag{3}$$

4 Algorithm of Resolution

By replacing the functions ψ_i and ϕ_i in the problem (1) by their values in the systems (2) and (3), we obtain:

$$\begin{cases} Min\ Ret(N_1, ..., N_n) = \dfrac{1}{W}\Bigg[\sum_{i=1}^{n}\sum_{t=1}^{M_i}\Big(S_i(1+V_i)(D_{it} - D_{it-1})\lambda_{it}N_i \\ \qquad + S_i(1+r)(\alpha_{it} - \alpha_{it-1})\mu_{it}N_i\Big) - \sum_{i=1}^{n} S_i(V_i - r)N_i \\ \qquad + \sum_{i=1}^{n}\sum_{t=1}^{M_i}(e_{it} - e_{it-1})\lambda_{it} + \sum_{i=1}^{n}\sum_{t=1}^{M_i}(1+r)(\beta_{it} - \beta_{it-1})\mu_{it}\Bigg] - r \\ \\ s.c. \quad N_i \le \dfrac{1}{2}\Bigg[\dfrac{\gamma_i W}{S_i} + \sum_{t=1}^{M_i}\left(\dfrac{a_{it} - a_{it-1}}{2S_i} - \dfrac{(\beta_{it} - \beta_{it-1})}{S_i}\right)\mu_{it} \\ \qquad + \sum_{t=1}^{M_i}\dfrac{(b_{it} - b_{it-1})}{2(1+V_i)S_i}\lambda_{it} - \sum_{t=1}^{M_i}(\alpha_{it} - \alpha_{it-1})N_i\mu_{it}\Bigg] \\ \\ \qquad N_i \ge \dfrac{1}{2}\Bigg[\dfrac{\delta_i W}{S_i} + \sum_{t=1}^{M_i}\left(\dfrac{a_{it-1} - a_{it-2}}{2S_i} - \dfrac{(\beta_{it} - \beta_{it-1})}{S_i}\right)\mu_{it} \\ \qquad + \sum_{t=1}^{M_i}\dfrac{(b_{it-1} - b_{it-2})}{2(1+V_i)S_i}\lambda_{it} - \sum_{t=1}^{M_i}(\alpha_{it} - \alpha_{it-1})N_i\mu_{it}\Bigg] \\ \\ \qquad \lambda_{it}, \quad \mu_{it} \in \{0,1\}: \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\ \qquad \lambda_{i0} \ge \lambda_{i1} \ge ... \ge \lambda_{iM_i}: \quad i \in \{1, ..., n\} \\ \qquad \mu_{i0} \ge \mu_{i1} \ge ... \ge \mu_{iM_i}: \quad i \in \{1, ..., n\} \\ \qquad a_{i(-1)} = a_{i0} = 0: \quad i \in \{1, ..., n\} \\ \qquad b_{i(-1)} = b_{i0} = 0: \quad i \in \{1, ..., n\}. \end{cases}$$

In order to linearize this problem, we pose $X_{it} = \lambda_{it} N_i$ and $Y_{it} = \mu_{it} N_i$. Then we replace them by their values and adding constraints in the problem above:

$$
\begin{cases}
Min \quad Ret(N_1, \ldots, N_n) = \dfrac{1}{W}\Big[\sum_{i=1}^{n}\sum_{t=1}^{M_i}\Big(S_i(1+V_i)(D_{it}-D_{it-1})X_{it} \\
\qquad +S_i(1+r)(\alpha_{it}-\alpha_{it-1})Y_{it}\Big) - \sum_{i=1}^{n} S_i(V_i - r)N_i \\
\qquad + \sum_{i=1}^{n}\sum_{t=1}^{M_i}(e_{it}-e_{it-1})\lambda_{it} + \sum_{i=1}^{n}\sum_{t=1}^{M_i}(1+r)(\beta_{it}-\beta_{it-1})\mu_{it}\Big] - r \\
s.c. \quad N_i \le \dfrac{1}{2}\Big[\dfrac{\gamma_i W}{S_i} + \sum_{t=1}^{M_i}\Big(\dfrac{a_{it}-a_{it-1}}{2S_i} - \dfrac{(\beta_{it}-\beta_{it-1})}{S_i}\Big)\mu_{it} \\
\qquad + \sum_{t=1}^{M_i}\dfrac{(b_{it}-b_{it-1})}{2(1+V_i)S_i}\lambda_{it} - \sum_{t=1}^{M_i}(\alpha_{it}-\alpha_{it-1})Y_{it}\Big] \\
\qquad N_i \ge \dfrac{1}{2}\Big[\dfrac{\delta_i W}{S_i} + \sum_{t=1}^{M_i}\Big(\dfrac{a_{it-1}-a_{it-2}}{2S_i} - \dfrac{(\beta_{it}-\beta_{it-1})}{S_i}\Big)\mu_{it} \\
\qquad + \sum_{t=1}^{M_i}\dfrac{(b_{it-1}-b_{it-2})}{2(1+V_i)S_i}\lambda_{it} - \sum_{t=1}^{M_i}(\alpha_{it}-\alpha_{it-1})Y_{it}\Big] \\
\qquad Y_{it} \le N_i : \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\
\qquad Y_{it} \le \mu_{it}\bar{N}_i : \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\
\qquad Y_{it} \ge \bar{N}_i\mu_{it} + N_i - \bar{N}_i : \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\
\qquad X_{it} \le N_i : \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\
\qquad X_{it} \le \lambda_{it}\bar{N}_i : \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\
\qquad X_{it} \ge \bar{N}_i\lambda_{it} + N_i - \bar{N}_i : \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\
\qquad \lambda_{it}, \quad \mu_{it} \in \{0, 1\} : \quad t \in \{0, ..., M_i\}, \quad i \in \{1, ..., n\} \\
\qquad \lambda_{i0} \ge \lambda_{i1} \ge ... \ge \lambda_{iM_i} : \quad i \in \{1, ..., n\} \\
\qquad \mu_{i0} \ge \mu_{i1} \ge ... \ge \mu_{iM_i} : \quad i \in \{1, ..., n\} \\
\qquad a_{i(-1)} = a_{i0} = 0 : \quad i \in \{1, ..., n\} \\
\qquad b_{i(-1)} = b_{i0} = 0 : \quad i \in \{1, ..., n\}.
\end{cases}
$$

5 Numerical Results and Conclusion

The formulation and its linearization have been implemented with Ilog Cplex 10.1. Numerical experiments were realized on the randomly generated instances.

These numerical results are presented in the following table:

	i=1	i=2	i=3	i=4	i=5
δ_i	0.012	0.013	0.011	0.012	0.015
γ_i	0.041	0.04	0.039	0.036	0.038
S_i	45	46	44	42	43
V_i	0.05	0.06	0.08	0.07	0.04
$Ret = 0.036$	$N_1 = 175$	$N_2 = 118$	$N_3 = 89$	$N_4 = 111$	$N_5 = 142$

	i=6	i=7	i=8	i=9	i=10
δ_i	0.014	0.011	0.011	0.013	0.0016
γ_i	0.048	0.047	0.039	0.046	0.047
S_i	46	47	49	44	44
V_i	0.05	0.09	0,049	0.054	0.06
$Ret = 0.036$	$N_6 = 176$	$N_7 = 128$	$N_8 = 213$	$N_9 = 164$	$N_{10} = 125$

	i=11	i=12	i=13	i=14	i=15
δ_i	0.002	0.012	0.0112	0.0132	0.011
γ_i	0.046	0.049	0.048	0.039	0.044
S_i	47	48	41	45	47
V_i	0.07	0.065	0.075	0.046	0.053
$Ret = 0.036$	$N_{11} = 108$	$N_{12} = 79$	$N_{13} = 101$	$N_{14} = 102$	$N_{15} = 106$

	i=16	i=17	i=18	i=19	i=20
δ_i	0.012	0.0115	0.0113	0.001	0.001
γ_i	0.038	0.042	0.049	0.03	0.036
S_i	49	47	46	42	57
V_i	0.091	0,0492	0.053	0.05	0.076
$Ret = 0.036$	$N_{16} = 98$	$N_{17} = 113$	$N_{18} = 74$	$N_{19} = 125$	$N_{20} = 108$

where ψ and ϕ are considered independent of i. They are taken as follows:

$$\psi(x) = \begin{cases} 0 & for \quad x = 0 \\ 11 & for \quad 0 < x \leq 3000 \\ 15 & for \quad 3000 < x \leq 7668 \\ 0.003x - 8 & for \quad 7668 < x \leq 8000 \\ 0.0055x - 23 & for \quad 8000 < x \leq 153000 \\ 0.004x - 23 & for \quad 153000 < x \leq 422000 \\ 0.0025x + 610 & for \quad x > 422000. \end{cases}$$

$$\phi(x) = \begin{cases} 0 & for \quad x = 0 \\ 12 & for \quad 0 < x \leq 4020 \\ 18 & for \quad 4020 < x \leq 8568 \\ 0.004x - 11 & for \quad 8568 < x \leq 12400 \\ 0.0068x - 25 & for \quad 12400 < x \leq 252000 \\ 0.0052x - 25 & for \quad 252000 < x \leq 543000 \\ 0.0032x + 840 & for \quad x > 543000. \end{cases}$$

The initial wealth is $W = 200000$. $r = 0.038$.

The results show that until 20 assets and 7 pieces in the affine functions, the linearized formulation runs very well. Beyond that, some improvements on the resolution scheme are needed. In this paper we treated the problem where the objective function is a combination of two functions of transaction. These last ones are discontinuous piecewise affine functions. We added binary variables to transform

the problem into an integer quadratique problem and afterwards linearized. This good theoretical result which consists in formulating the problem in the form of an integer linear program allowed us to resolve him by using the ILOG software. The results of numerical simulations are encouraging and prove the efficiency of our new approach. As perspectives, such improvements may be reached by strengthening the linear formulation polytope with cuts, reducing the size of the variables N_i, or applying different approaches to the integer quadratic formulation such as semidefinite programming.

References

1. Akian, M., Menaldi, J.L., Sulem, A.: On an Investment-Consumption Model with Transaction Costs. SIAM Journal of Control and Optimization 34(1), 329–364 (1996)
2. Dumas, B., Luciano, E.: An Exact Solution to a Dynamic Portfolio Choice Problem under Transaction Costs. Journal of Finance XLVI (2), 577–595 (1991)
3. Davis, M.H.A., Norman, A.R.: Portfolio Selection with Transaction Costs. Mathematics of Operations Research 15(4), 676–713 (1990)
4. Shreve, S., Soner, H.M.: Optimal Investment and Consumption with Transaction Costs. Annals of Applied Probability 4, 609–692 (1994)
5. Eastham, J., Hastings, K.: Optimal Impulse Control of Portfolios. Mathematics of Operations Research (13), 588–605 (1988)
6. Mansini, M., Speranza, M.: Heuristic algorithms for the portfolio selection problem with minimum transaction lots. European Journal of Operational Research (114), 219–233 (1999)
7. George, L.N.: Models for representing piecewise linear cost functions. Operations Research Letters (32) 44–48 (2004)
8. Gabrel, V., Knippel, A., Minoux, M.: Exact solution of multicommodity network optimization problems with general step cost functions. Operations Research Letters 25, 15–23 (1999)
9. Fortet, R.: Applications de l'algèbre de boole en recherche opŕationnelle. Revue Française d'Automatique, d'Informatique et de Recherche Opérationnelle 4, 5–36 (1959)
10. Souad, L.-J., Rakotondratsimba, Y.: The number of securities giving the maximum return in the presence of transaction costs. Quality and Quantity (2007), Doi:10.1007/s11135–007–9126–y
11. Markowitz, H.M.: Portfolio Selection. Journal of Finance 7, 77–91 (1952)

An Exact Method for a Discrete Quadratic Fractional Maximum Problem

Nacéra Maachou[1] and Mustapha Moulaï[2]

[1] LAID3, Faculty of Mathematics, USTHB BP 32, Bab-Ezzouar 16111, Algeria
nacera_maachou@yahoo.fr
[2] Laboratoire LAID3, Département de Recherche Opérationnelle,
USTHB BP. 32, EL ALIA 16111 Alger, Algérie

Abstract. In this work, we develop a new algorithm for solving a discrete quadratic fractional maximum problem in which the objective is to optimize a ratio of two quadratic functions over a set of integer points contained in a convex polytope. This algorithm is based on a branch and bound method on computation of penalties and a related integer linear fractional programs.For this problem, optimality conditions are derived. A numerical example is presented for illustrating the proposed method.

1 Introduction

Fractional Programming problems have been a subject of wide interest since they arise in many fields like agricultural planning, financial analysis of a firm, location theory, capital budgeting problem, portfolio selection problem, cutting stock problem, stochastic process problem. From time to time survey papers on applications and algorithms on fractional programming have been presented by many authors [3,5,6,7]. In this paper a new algorithm is developed for solving integer quadratic fractional programs in which the objective is to minimize a ratio of two quadratic functionals over a set of integer points contained in a convex polytope. Branch and bound method based on computation of penalties is developed to solve the problem. The branch and bound methods have been used in literature for solving a number of integer programming problems. For solving the problem, integer points of the polytope are ranked in non-decreasing order of the values of the integer quadratic fractional programming problem. Integer linear fractional programming problem related to the main problem is formulated and its feasible integer solutions are scanned in a systematic manner till an integer optimal solution of the problem is obtained.

2 Notations and Definitions

Let S the set of vectors $x \in \mathbb{R}^n$ satisfying the constraints $x \geq 0$, x integer and $Ax \leq b$ where A is an integer $m \times n$ matrix and b a vector of $\mathbb{R}^m$. Let C, D vectors of $\mathbb{R}^n$, E, F are real symmetric matrices and α, β two elements of $\mathbb{R}$.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 197–203, 2008.

The integer quadratic fractional programming problem (P), intended to be studied can be mathematically stated as:

$$(P): \ \min f(x) = \frac{C^T x + x^T E x + \alpha}{D^T x + x^T F x + \beta}, \ x \in S \ . \tag{1}$$

$D^T x + x^T F x + \beta > 0$ for all $x \in \widetilde{S} = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$ where the nonempty $\widetilde{S}$ is bounded.

To find an optimal integer solution of problem (P), integer feasible solutions are ranked in non-decreasing order of the values of the objective function. For obtaining various integer feasible solutions a related integer linear fractional programming problem (P_1) is constructed.

$$(P_1): \ \min g(x) = \frac{U^T x + \alpha}{V^T x + \beta}, \ x \in S \ . \tag{2}$$

where

$U_j =$ j^{th} component of $U \in \mathbb{R}^n = \min\limits_{x \in \widetilde{S}} \ (C_j + x^T E_j), j = 1, ..., n$, C_j being the j^{th} column of C and E_j being the j^{th} column of E.

$V_j =$ j^{th} component of $V \in \mathbb{R}^n = \max\limits_{x \in \widetilde{S}} (D_j + x^T F_j), j = 1, ..., n$, D_j being the j^{th} column of D and F_j being the j^{th} column of F.

3 Some Results

Proposition 1

$$g(x) \leq f(x), \ \forall x \in \widetilde{S} \ . \tag{3}$$

Proof. By definition of U_j and V_j, we have:

$\forall x \in \widetilde{S}$, $U_j \leq C_j + x^T E_j$ and $V_j \geq D_j + x^T F_j, \quad j = 1, ..., n$.

As $x \geq 0$, $\forall x \in \widetilde{S} \quad U^T x \leq (C^T + x^T E)x$ and $V^T x \geq (D^T + x^T F)x$

Clearly it follows that $\forall x \in \widetilde{S}, \ g(x) \leq f(x)$. □

Recall that the integer linear fractional problem (P_1) can be solved by branch and bound method based on computation of penalties [1]. Consider the problem (P_2) without integrality restrictions

$$(P_2): \ \min g(x) = \frac{U^T x + \alpha}{V^T x + \beta}, \ x \in \widetilde{S} \ . \tag{4}$$

and introducing slack variables and solving it by Cambini and Martein's method [4], we find the optimal continuous solution. Let the optimal simplex tableau be given by (P_2'):

$$(P_2'): \ \min g(x) = \frac{\bar{\alpha} + \sum\limits_{j \in I_N}^{p_j} x_j}{\bar{\beta} + \sum\limits_{j \in I_N}^{q_j} x_j} \ . \tag{5}$$

$$x_i + \sum_{j \in I_N} \bar{a}_{ij} x_j = e_i, \quad i \in I_B,\ x_j \geq 0,\ j \in I_N \ . \tag{6}$$

where I_N is the index set of the non-basic variables, $x_i,\ i \in I_B$ is the basic variable, $\bar{\alpha}$ and $\bar{\beta}$ are the reduced costs in the simplex tableau and $\bar{\alpha}/\bar{\beta}$ is the value of the objective function. The optimal basic solution of problem (P_2) is given by

$$x_i = e_i, \quad i \in I_B \text{ otherwise } x_j = 0 \quad j \in I_N \quad . \tag{7}$$

If e_i is integer for every $i \in I_B$, then the integer optimal solution to problem (P_2) is obtained. If the necessary integrality restrictions are not satisfied, let e_{kt} be non integer value of x_{kt} for some $kt \in I_B$. We denote the largest integer less than e_{kt} by $\lfloor e_{kt} \rfloor$ and the smallest integer greater than e_{kt} by $\lceil e_{kt} \rceil$. Since x_{kt} is required to be integer, either $x_{kt} \leq \lfloor e_{kt} \rfloor$ or $x_{kt} \geq \lceil e_{kt} \rceil$.

Let us consider the former $x_{kt} \leq \lfloor e_{kt} \rfloor$which gives rise to the constraint $x_{_{kt}} + s = \lfloor e_{kt} \rfloor$ but $x_{_{kt}} + \sum\limits_{j \in I_N} \bar{a}_{ktj} x_j = e_{kt}$ from the simplex tableau. Then

$$-\sum_{j \in I_N} \bar{a}_{ktj} x_j + s = \lfloor e_{kt} \rfloor - e_{kt} \ . \tag{8}$$

Thus we have

$$-\sum_{j \in I_N} \bar{a}_{ktj} x_j \leq \lfloor e_{kt} \rfloor - e_{kt} \ . \tag{9}$$

It is obvious that $\lfloor e_{kt} \rfloor - e_{kt}$ is negative and the optimal solution to problem (P_1) given below does not satisfy the constraint (9). Augmenting this constraints to problem (P_1), we obtain one of the branches.

Similarly, corresponding to $x_{kt} \geq \lceil e_{kt} \rceil$, we obtain the constraint

$$-x_{kt} + s = -\lceil e_{kt} \rceil \ . \tag{10}$$

Then

$$\sum_{j \in I_N} \bar{a}_{ktj} x_j \leq e_{kt} - \lceil e_{kt} \rceil < 0 \ . \tag{11}$$

Introducing this constraints to problem (P_1), we obtain the other branch. For selecting a branch which must be added to the optimal simplex tableau (P_2'), we compute the penalties π_r and π_r' of the constraints $x_{kt} \leq \lfloor e_{kt} \rfloor$ and $x_{kt} \geq \lceil e_{kt} \rceil$, respectively given by:

$$\pi_r = \frac{e\Delta_r}{\bar{\beta}(\bar{\beta} + \frac{e\bar{q}_r}{\bar{a}_{kt,r}})} \ . \tag{12}$$

and

$$\pi_r' = \frac{(1-e)\Delta_r'}{\bar{\beta}(\bar{\beta} + \frac{(1-e)\bar{q}_r}{\bar{a}_{kt,r'}})} \ . \tag{13}$$

where

$$\Delta_r = \min\left\{ \frac{\bar{\gamma}_j}{-\bar{a}_{kt,j}} \mid \bar{a}_{kt,j} > 0 \right\}, \quad \Delta_r' = \min\left\{ \frac{\bar{\gamma}_j}{\bar{a}_{kt,j}} \mid \bar{a}_{kt,j} < 0 \right\} \ . \tag{14}$$

and $e = e_{kt} - \lfloor e_{kt} \rfloor$.

γ_j represents the j^{th} component of the reduced gradient of g at x, $\bar{\gamma}_j = \bar{\beta} \times \bar{p}_j - \bar{\alpha} \times \bar{q}_j, \quad j \in I_N$.

The branch corresponding to the smallest penalty is augmented to problem (P_2').

Proposition 2. *If for some $k \geq 1$, $g_k \geq \min\{f(x) \mid x \in T^k\} = f(\hat{x})$, then $\hat{x}$ is the optimal solution of (P).*

Proof. We have

$$\forall x \in T^k \quad f(\hat{x}) \leq f(x) \ . \tag{15}$$

On the other hand from Proposition 1 and our hypothesis it follows that

$$\forall x \in S/T^k \quad f(x) \geq g(x) \geq g_{k+1} > g_k \geq f(\hat{x}) \ . \tag{16}$$

Conditions (15) and (16) implies that $\hat{x}$ is the optimal solution of (P). □

Next proposition shows that when the hypothesis of Proposition 2 is not satisfied, we have information about the minimum value of the initial problem (P).

Proposition 3. *If $g_k < \min\{f(x) \mid x \in T^k\}$, then $g_k < f_1 \leq \min\{f(x) \mid x \in T^{k+1}\}$.*

Proof. The second inequality is obvious. For the first part, we note as in proposition 2 that

$$\forall x \in S/T^k \quad f(x) \geq g(x) \geq g_{k+1} > g_k \ . \tag{17}$$

Therefore

$$\min\{f(x) : x \in S/T^k\} > g_k \ . \tag{18}$$

By hypothesis,

$$\min\{f(x) : x \in T^k\} > g_k \ . \tag{19}$$

Then conditions (18) and (19) imply that $f_1 > g_k$. □

4 Notations

Δ_i =Set of all the i^{th}best feasible solutions of (P_1);

Clearly

Δ_1 =Set of optimal solutions of (P_1),g_1 the optimal value corresponding at Δ_1 and

Δ_2 = Set of the 2^{nd} optimal solutions of (P_1), g_2 the optimal value corresponding at Δ_2. It follows that $g_1 < g_2$. Obviously for $i = k$ we have $g_k < g_{k+1}$ and the feasible set S being finite. Introduce the notations $T^k = \Delta_1 \cup \Delta_2 \cup ... \cup \Delta_k$.

4.1 Algorithm

Step 0. Find $C_j = \min\limits_{x\in\widetilde{S}} (C_j + x^T E_j)$ and $V_j = \mathop{Max}\limits_{x\in\widetilde{S}} (D_j + x^T F_j), \quad j = 1...n.$ Construct the related integer linear fractional programming problem (P_1) and go to step 1.

Step 1. Solve problem (P_2) .

- if a such solution does not exist, stop. Either $\sup\limits_{x\in S} g = +\infty$
- Otherwise, set $k = 1, l = 1$ and goto step2.

Step 2. Let x^1 an optimal continuous solution of problem (P_2).

- if x^1 is integer, Find Δ_1 and compute $g(x^1) = g_1$, go to step 5.
- Otherwise for $k = 1, \ \ l = 1$, let x_{kt} be for some $kt \in I_B$, a non integer component of x^1 with corresponding value e_{kt}. Set $\pi_l = 0$ and go to step 3.

Step 3. Compute π_{2k-1} and π_{2k}. Let $\pi_{2k-1} = \pi_{2k-1} + \pi_l, \pi_{2k} = \pi_{2k} + \pi_l$ and $\pi_l = +\infty$. Compute $\pi_l = \min\limits_{1\leq j\leq 2k} \{\pi_j\}$. Augment the constraint to the optimal simplex tableau, solve it and go to step 4.

Step 4. if x^l is integer, Find Δ_l and compute $g(x^l) = g_l$, go to step 7. Otherwise, the augmented problems have no solutions, stop. Let x^l_{kt} be a non integer component of x^l with corresponding value e^l_{kt}. Set $k = k + 1$ and go to step 3.

Step 5. Test

- if $g_1 = \min\left\{f(x) \mid x \in T^1\right\} = f(\hat{x}_1)$. Then $\hat{x}_1$ is the integer optimal solution of problem (P).
- If $g_1 < \min\left\{f(x) \mid x \in T^1\right\}$, $l = 2$ and go to step 6.

Step 6. Find the next best solution of the problem(P_2), go to step 4.

Step 7. Test

- If $g_l \geq \min\left\{f(x) \mid x \in T^l\right\} = f(\hat{x}_1)$, then $\hat{x}_1$ is the integer optimal solution of problem (P).
- If $g_l < \min\left\{f(x) \mid x \in T^l\right\}$, $l = l + 1$ and go to step 6.

5 Illustrative Example

Consider the integer quadratic fractional programming problem (P):

$$(P): \ \min \quad f(x_1, x_2) = \frac{-5x_1 - 8x_2 - x_1^2 - 2x_1x_2 - 3}{2x_1 + x_1^2 + 2} \ . \tag{20}$$

$$(x_1, x_2) \in S = \{2x_1 + 3x_2 \leq 6; \ 3x_1 + 2x_2 \leq 5; \ x_1 \geq 0; \ x_2 \geq 0, integers\} \ . \tag{21}$$

Step 0: A related integer linear fractional problem (P_1) is constructed as:

$$(P_1): \ \min \quad f(x_1, x_2) = \frac{-7x_1 - 9x_2 - 3}{3x_1 + 4x_2 + 2}, \ (x_1, x_2) \in S \ . \tag{22}$$

For solving problem (P_1), a relaxed linear fractional programming problem (P_2) say is given by:

$$(P_1): \min \quad f(x_1, x_2) = \frac{-7x_1 - 9x_2 - 3}{3x_1 + 4x_2 + 2},\ (x_1, x_2) \in \widetilde{S}\ . \tag{23}$$

Step 1: The optimal continuous solution is $x_1 = 3/5,\ x_2 = 8/5$ and $x_j = 0$ otherwise.

Step 2: Since x_1 and x_2 are not integers, this is not the one corresponding to the required solution to problem (P_1).

Step 3: Compute the penalties π_1 and π_2 of the added constraints $x_1 \leq \lfloor 3/5 \rfloor$ and $x_1 \geq \lceil 3/5 \rceil$, respectively, we obtain $\pi_1 = 3/170$ and $\pi_2 = 1/153$. Then, we select the branch whose penalty is smallest and augment respective constraint to the optimal simplex tableau (P_2').

Step 4: The solution that we obtain is $x_1 = 1,\ x_2 = 1,\ x_3 = 1$ and $x_j = 0$ otherwise. $\Delta_1 = \{(1, 1)\}$ and $g_1 = 19/9$.

Step 5: $\min\left\{f(x) \mid x \in T^1\right\} = 19/9$**.** Then $g_1 = \min\left\{f(x) \mid x \in T^1\right\} = f(\hat{x}_1)$. Then $\hat{x}_1$ is the integer optimal solution of problem (P).

6 Conclusion

In this paper, we have proposed an algorithm for solving integer quadratic fractional programs. This algorithm is based on a branch and bound method on computation of penalties and a related integer linear fractional programming problem is constructed for solving the integer quadratic fractional programs problem in a finite number of iterations. we hope that this article motivates the researchers to develop better solution procedures for this problem.

Acknowledgements

The authors are grateful to anonymous referees for their substantive comments that improved the content and presentation of the paper. This research has been completely supported by the laboratory LAID3 of the High Education Algerian Ministry.

References

1. Abbas, M., Moulaï, M.: Penalties Method for Integer Linear Fractional Programs. Jorbel 37, 41–51 (1997)
2. Abbas, M., Moulaï, M.: An algorithm for mixed integer linear fractional programming problem. Jorbel 39, 21–30 (1999)
3. Abbas, M., Moulaï, M.: Integer Linear Fractional Programming with Multiple Objective. Ricerca Operativa, 103–104, 51–70 (2002)

4. Cambini, A., Martein, L.: Equivalence in Linear Fractional Programming. Optimization 23, 41–51 (1992)
5. Craven, B.D.: Fractional programming. Sigma Series in Applied Mathematics, vol. 4. Heldermann Verlag (1988)
6. Sniedovich., M.: Fractional programming revisited. EJOR 33, 334–341 (1998)
7. Stancu-Minasian, I.M.: A sixth bibliography of fractional programming. Optimization 55, 405–428 (2006)

Disaggregation of Bipolar-Valued Outranking Relations

Patrick Meyer[1], Jean-Luc Marichal[2], and Raymond Bisdorff[2]

[1] Institut TELECOM, TELECOM Bretagne, LabSTICC - UMR 3192,
Technople Brest-Iroise CS 83818, F-29238 Brest Cedex 3, France
patrick.meyer@telecom-bretagne.eu
[2] University of Luxembourg, 162a, avenue de la Faïencerie, L-1511 Luxembourg
raymond.bisdorff, jean-luc.marichal@uni.lu

Abstract. In this article, we tackle the problem of exploring the structure of the data which is underlying a bipolar-valued outranking relation. More precisely, we show how the performances of alternatives and weights related to criteria can be determined from three different formulations of the bipolar-valued outranking relations, which are given beforehand.

1 Introduction

Let $X = \{x, y, z, \ldots\}$ be a set of p alternatives and $N = \{1, \ldots, n\}$ be a set of n criteria. Each alternative of X is evaluated on each of the criteria of N. Let us write $g_i(x)$ for the performance of alternative x on criterion i of N. In this work, we will regard, without any loss of generality, such a *performance function* g_i $(i \in N)$ as having its values in $[0, 1]$ s.t.:

$$\forall x, y \in X,\ g_i(x) \geq g_i(y) \Rightarrow x \text{ is } \textit{at least as good as } y \text{ on criterion } i. \quad (1)$$

With each criterion i of N we associate its *weight* represented by a rational number w_i from the interval $[0, 1]$ such that

$$\sum_{i=1}^{n} w_i = 1.$$

To enrich the model which can be based on Formula (1), it is possible to associate different thresholds (weak preference, preference, weak veto, veto; see, e.g., [1]) with the criteria functions which allow to represent more precisely a decision maker's (DM's) local "*at least as good as*" preferences.

Let S be a binary relation on X. Classically, the proposition "x *outranks* y" (xSy) $(x, y \in X)$ is assumed to be validated if there is a sufficient majority of criteria which supports an "*at least as good as*" preferential statement and there is no criterion which raises a *veto* against it [2].

In this paper, given the outranking relation, we detail how the performances of the alternatives and the weights associated with the criteria can be determined. We present three different definitions of the outranking relation, where the first

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 204–213, 2008.

model takes only into account a preference threshold, the second one considers also a weak preference threshold, and finally, the third one adds also two veto thresholds.

From a practical point of view, the determination of the performances of the alternatives on the criteria may be questionnable, as in general, in a decision problem, these evaluations are given beforehand. Nevertheless, from an experimental point of view, the determination of a performance table from a given valued outranking relation can be of some help. Furthermore, it is possible to show that our developments can easily be extended to the tuning of the parameters underlying the DM's preferences.

2 $\mathcal{M}_1$: Model with a Single Preference Threshold

Starting from Formula (1), this first model enriches the local pairwise comparison of two alternatives on each criterion by a preference threshold. Therefore, to characterise a local "*at least as good as*" situation between two alternatives x and y of X, for each criterion i of N, we use the function $C_i : X \times X \rightarrow \{-1, 1\}$ defined by:

$$C_i(x,y) = \begin{cases} 1 & \text{if} \quad g_i(y) < g_i(x) + p\,; \\ -1 & \text{otherwise}\,, \end{cases} \tag{2}$$

where $p \in]0,1[$ is a constant preference threshold associated with all the preference dimensions. According to this local concordance index, x is considered as *at least as good as* y for criterion i if $g_i(y) < g_i(x) + p$ ($C_i(x,y) = 1$). Else, x is not considered as at least as good as y for criterion i ($C_i(x,y) = -1$).

The overall outranking index $\widetilde{S}$, defined for all pairs of alternatives $(x,y) \in X \times X$, can then be written as:

$$\widetilde{S}(x,y) = \sum_{i \in N} w_i C_i(x,y). \tag{3}$$

$\widetilde{S}$ represents the *credibility of the validation or non-validation* of an outranking situation observed between each pair of alternatives [1]. The maximum value 1 of $\widetilde{S}$ is reached in the case of *unanimous concordance*, whereas the minimum value -1 is obtained in the case of *unanimous discordance*. $\widetilde{S}$ is called the *bipolar-valued characterisation* of the outranking relation S, or, for short, the *bipolar-valued outranking relation*.

Given the bipolar-valued outranking relation $\widetilde{S}$ and a constant preference threshold p, we now show how the values taken by the performance functions $g_i(x)$ ($\forall i \in N, \forall x \in X$) and the associated weights w_i ($\forall i \in N$) can be determined.

The local concordance conditions (2) can be translated as follows into linear constraints:

$$(-1+p)(C_i(x,y)-1) < g_i(x) - g_i(y) + p \leq (1+p)(C_i(x,y)+1) \quad \forall x \neq y \in X, \forall i \in N, \tag{4}$$

where $C_i(x,y) \in \{-1,1\}$ for each $x \neq y \in X$. Indeed, $g_i(x) - g_i(y) + p > 0$ implies $C_i(x,y) = 1$ whereas $g_i(x) - g_i(y) + p \leq 0$ forces $C_i(x,y) = -1$.

Constraints derived from Equation 3 can be written as

$$\sum_{i=1}^{n} w'_i(x,y) = \widetilde{S}(x,y) \quad \forall x \neq y \in X,$$

where $w'_i(x,y)$ is a non-negative variable for each $i \in N$, $x \neq y \in X$ s.t.:

$$w'_i(x,y) = \begin{cases} w_i \text{ if } C_i(x,y) = 1; \\ -w_i \text{ otherwise.} \end{cases}$$

This then leads to the following linear constraints:

$$\begin{array}{l} -w_i \leq w'_i(x,y) \leq w_i; \\ w_i + C_i(x,y) - 1 \leq w'_i(x,y); \\ w'_i(x,y) \leq -w_i + C_i(x,y) + 1. \end{array}$$

Indeed, $C_i(x,y) = -1$ implies $w'_i(x,y) = -w_i$, whereas $C_i(x,y) = 1$ forces $w'_i(x,y) = w_i$.

In order to remain flexible enough and not to depend on rounding errors, we propose to approach the values taken by $\widetilde{S}$ as closely as possible by minimising the maximal gap between $\widetilde{S}(x,y)$ and $\sum_{i \in N} w_i C_i(x,y)$, for all $x \neq y \in X$, represented by a non-negative variable ε.

The mixed integer program **MIP1** which has to be solved can now be written as follows:

MIP1:

Variables:

$\varepsilon \geq 0$	
$g_i(x) \in [0,1]$	$\forall i \in N, \forall x \in X$
$w_i \in [0,1]$	$\forall i \in N$
$C_i(x,y) \in \{0,1\}$	$\forall i \in N, \forall x \neq y \in X$
$w'_i \in [-1,1]$	$\forall i \in N$

Parameters:

$p \in]0,1[$	
$\widetilde{S}(x,y) \in [0,1]$	$\forall x \neq y \in X$
$\delta \in]0,p[$	

Objective function:

min ε

Constraints:

s.t.	$\sum_{i=1}^{n} w_i = 1$	
	$-w_i \leq w'_i(x,y)$	$\forall x \neq y \in X, \forall i \in N$
	$w'_i(x,y) \leq w_i$	$\forall x \neq y \in X, \forall i \in N$
	$w_i + C_i(x,y) - 1 \leq w'_i(x,y)$	$\forall x \neq y \in X, \forall i \in N$

$$
\begin{array}{ll}
w'_i(x,y) \leq -w_i + C_i(x,y) + 1 & \forall x \neq y \in X, \forall i \in N \\
\sum_{i=1}^{n} w'_i(x,y) \leq \widetilde{S}(x,y) + \varepsilon & \forall x \neq y \in X \\
\sum_{i=1}^{n} w'_i(x,y) \geq \widetilde{S}(x,y) - \varepsilon & \forall x \neq y \in X \\
(-1+p)(1 - C_i(x,y)) + \delta \leq g_i(x) - g_i(y) + p & \forall x \neq y \in X, \forall i \in N \\
g_i(x) - g_i(y) + p \leq (1+p)C_i(x,y) & \forall x \neq y \in X, \forall i \in N
\end{array}
$$

The solution of **MIP1** might not be unique. If the objective function equals 0, then there exist $g_i(x)$ $(\forall i \in N, \forall x \in X)$ and associated weights w_i $(\forall i \in N)$ generating the overall outranking index $\widetilde{S}$ via Equations (2) and (3). Else there exists no solution to the problem via the selected representation, and the output of **MIP1** can be considered as an approximation of the given $\widetilde{S}$ by a the constant preference threshold model.

Let us now turn to a more complex model which allows to represent a larger set of valued outranking relations.

3 $\mathcal{M}_2$: Model with Two Preference Thresholds

In this case, a local "*at least as good as*" situation between two alternatives x and y of X is characterised by the function $C'_i : X \times X \to \{-1, 0, 1\}$ s.t.:

$$
C'_i(x,y) = \begin{cases} 1 & \text{if} \quad g_i(y) < g_i(x) + q\,; \\ -1 & \text{if} \quad g_i(y) \geq g_i(x) + p\,; \\ 0 & \text{otherwise}\,, \end{cases} \tag{5}
$$

where $q \in]0, p[$ is a constant weak preference threshold associated with all the preference dimensions. If $C'_i(x,y) = 1$ (resp. $C'_i(x,y) = -1$), then x is considered (resp. not considered) as *at least as good as* y for criterion i. Finally, according to the developments in [1], if $g_i(x) + q \leq g_i(y) < g_i(x) + p$ then it cannot be determined whether x is *at least as good as* y or not for criterion i, and $C'_i(x,y) = 0$.

The overall outranking index $\widetilde{S}'$ is defined as follows for all pairs of alternatives $(x,y) \in X \times X$:

$$
\widetilde{S}'(x,y) = \sum_{i \in N} w_i C'_i(x,y). \tag{6}
$$

According to Equation (6), $\widetilde{S}'$ has its values in $[-1, 1]$. Its maximum value 1 is reached in the case of *unanimous concordance*, its minimum value -1 represents *unanimous discordance*, and the value 0 is obtained if the positive arguments counterbalance the negative arguments for the outranking. The value 0 therefore represents an *indetermined* outranking situation. In this context, $\widetilde{S}'$ is again called the *bipolar-valued* outranking relation.

In order to represent the three values taken by $C'_i(x,y)$, we use two binary variables $\alpha_i(x,y) \in \{0,1\}$ and $\beta_i(x,y) \in \{0,1\}$ $(\forall i \in N, \forall x \neq y \in X)$ s.t.

$$
C'_i(x,y) = \alpha_i(x,y) - \beta_i(x,y). \tag{7}
$$

Note that $C'_i(x,y) = 1$ if $\alpha_i(x,y) = 1$ and $\beta_i(x,y) = 0$, $C'_i(x,y) = -1$ if $\alpha_i(x,y) = 0$ and $\beta_i(x,y) = 1$, and $C'_i(x,y) = 0$ if $\alpha_i(x,y) = \beta_i(x,y) = 1$ or $\alpha_i(x,y) = \beta_i(x,y) = 0$.

The local concordance conditions (5) can then be rewritten as follows as linear constraints ($\forall x \neq y \in X, \forall i \in N$):

$$\begin{cases} (-1+q)(1-\alpha_i(x,y)) < g_i(x) - g_i(y) + q \leq (1+q)\alpha_i(x,y); \\ (-1+p)\beta_i(x,y) < g_i(x) - g_i(y) + p \leq (1+p)(1-\beta_i(x,y)). \end{cases} \tag{8}$$

Note that, as $p > q > 0$, $g_i(x) - g_i(y) + q > 0 \Rightarrow g_i(x) - g_i(y) + p > 0$, and $g_i(x) - g_i(y) + p < 0 \Rightarrow g_i(x) - g_i(y) + q < 0$. Consequently, in constraints (8), $g_i(x) - g_i(y) + q > 0$ forces $\alpha_i(x,y) = 1$ and $\beta_i(x,y) = 0$ ($C'_i(x,y) = 1$) whereas $g_i(x) - g_i(y) + p < 0$ implies $\beta_i(x,y) = 1$ and $\alpha_i(x,y) = 0$ ($C'_i(x,y) = -1$). Furthermore, $g_i(x) - g_i(y) + q < 0$ and $g_i(x) - g_i(y) + p > 0$ implies $\alpha_i(x,y) = \beta_i(x,y) = 0$ ($C'_i(x,y) = 0$). Then, $g_i(x) - g_i(y) + q = 0 \Rightarrow g_i(x) - g_i(y) + p > 0$ forces $\alpha_i(x,y) = \beta_i(x,y) = 0$ and finally $g_i(x) - g_i(y) + p = 0 \Rightarrow g_i(x) - g_i(y) + q < 0$ implies that $\alpha_i(x,y) = 0$ and $\beta_i(x,y) = 1$ ($C'_i(x,y) = 1$).

It is important to note that constraints (8) linked to the condition $p > q > 0$ do not allow that $\alpha_i(x,y) = \beta_i(x,y) = 1$ simultanously. Indeed $\alpha_i(x,y) = 1 \Rightarrow g_i(x) - g_i(y) + q \geq 0$ and $\beta_i(x,y) = 1 \Rightarrow g_i(x) - g_i(y) + p \leq 0$, which is only possible if $p = q$.

Equation (6) can be rewritten as follows:

$$\widetilde{S}'(x,y) = \sum_{i \in N} w_i(\alpha_i(x,y) - \beta_i(x,y)) \quad \forall x \neq y \in X,$$

which can be replaced by a linear constraint of the type

$$\sum_{i=1}^{n} w''_i(x,y) = \widetilde{S}'(x,y) \quad \forall x \neq y \in X,$$

where $w''_i(x,y) \in [-1,1]$ for each $i \in N, x \neq y \in X$ s.t.:

$$w''_i(x,y) = \begin{cases} w_i \text{ if } C'_i(x,y) = 1; \\ -w_i \text{ if } C'_i(x,y) = -1; \\ 0 \text{ otherwise.} \end{cases}$$

This then leads to the following linear constraints ($\forall x \neq y \in X, \forall i \in N$):

$$\begin{aligned} &-w_i \leq w''_i(x,y) \leq w_i; \\ &w_i + \alpha_i(x,y) - \beta_i(x,y) - 1 \leq w''_i(x,y) \\ &w''_i(x,y) \leq -w_i + \alpha_i(x,y) - \beta_i(x,y) + 1 \\ &-[\alpha_i(x,y) + \beta_i(x,y)] \leq w''_i(x,y) \leq \alpha_i(x,y) + \beta_i(x,y). \end{aligned}$$

Indeed, recalling that $\alpha_i(x,y)$ and $\beta_i(x,y)$ cannot simultanously be equal to 1, it is easy to verify that $C'_i(x,y) = 1 \Rightarrow w''_i(x,y) = w_i$, $C'_i(x,y) = -1 \Rightarrow w''_i(x,y) = -w_i$, and $C'_i(x,y) = 0 \Rightarrow w''_i(x,y) = 0$.

These considerations lead to the formulation of the mixed integer program **MIP2**, whose objective is again to minimise a non-negative variable ε representing the maximal gap between $\widetilde{S}'(x,y)$ and $\sum_{i\in N} w_i C'_i(x,y)$, for all $x \neq y \in X$.

MIP2:

Variables:

$\varepsilon \geq 0$	
$g_i(x) \in [0,1]$	$\forall i \in N, \forall x \in X$
$w_i \in [0,1]$	$\forall i \in N$
$\alpha_i(x,y) \in \{0,1\}$	$\forall i \in N, \forall x \neq y \in X$
$\beta_i(x,y) \in \{0,1\}$	$\forall i \in N, \forall x \neq y \in X$
$w''_i(x,y) \in [-1,1]$	$\forall i \in N, \forall x \neq y \in X$

Parameters:

$q \in]0,p[$	
$p \in]q,1[$	
$\widetilde{S}'(x,y) \in [0,1]$	$\forall x \neq y \in X$
$\delta \in]0,q[$	

Objective function:

min ε

Constraints:

s.t. $\sum_{i=1}^{n} w_i = 1$	
$-w_i \leq w''_i(x,y)$	$\forall x \neq y \in X, \forall i \in N$
$w''_i(x,y) \leq w_i$	$\forall x \neq y \in X, \forall i \in N$
$w_i + \alpha_i(x,y) - \beta_i(x,y) - 1 \leq w''_i(x,y)$	$\forall x \neq y \in X, \forall i \in N$
$w''_i(x,y) \leq -w_i + \alpha_i(x,y) - \beta_i(x,y) + 1$	$\forall x \neq y \in X, \forall i \in N$
$-[\alpha_i(x,y) + \beta_i(x,y)] \leq w''_i(x,y)$	$\forall x \neq y \in X, \forall i \in N$
$w''_i(x,y) \leq \alpha_i(x,y) + \beta_i(x,y)$	$\forall x \neq y \in X, \forall i \in N$
$\sum_{i=1}^{n} w''_i(x,y) \leq \widetilde{S}'(x,y) + \varepsilon$	$\forall x \neq y \in X$
$\sum_{i=1}^{n} w''_i(x,y) \geq \widetilde{S}'(x,y) - \varepsilon$	$\forall x \neq y \in X$
$(-1+q)(1-\alpha_i(x,y)) + \delta \leq g_i(x) - g_i(y) + q$	$\forall x \neq y \in X, \forall i \in N$
$g_i(x) - g_i(y) + q \leq (1+q)\alpha_i(x,y)$	$\forall x \neq y \in X, \forall i \in N$
$(-1+p)\beta_i(x,y) + \delta \leq g_i(x) - g_i(y) + p$	$\forall x \neq y \in X, \forall i \in N$
$g_i(x) - g_i(y) + p \leq (1+p)(1-\beta_i(x,y))$	$\forall x \neq y \in X, \forall i \in N$

Similar remarks as for **MIP1** concerning the uniqueness and the characteristics of the solution apply here. Once again, let us now turn to a more complex model which allows to represent an even larger set of valued outranking relations.

4 $\mathcal{M}_3$: Model with Two Preference and Two Veto Thresholds

In this third case, the outranking relation is enriched by veto thresholds on the criteria. A veto threshold on a criterion $i \in N$ allows to clearly non-validate an outranking situation between two alternatives if the difference of evaluations on i is too large. A *local veto* situation for each criterion i of N is characterised by a veto function $V_i : X \times X \to \{-1, 0, 1\}$ s.t.:

$$V_i(x,y) = \begin{cases} 1 & \text{if} \quad g_i(y) \geq g_i(x) + v\,; \\ -1 & \text{if} \quad g_i(y) < g_i(x) + wv\,; \\ 0 & \text{otherwise}\,, \end{cases} \tag{9}$$

where $wv \in]p, 1[$ (resp. $v \in]wv, 1[$) is a constant weak veto threshold (resp. veto threshold) associated with all the preference dimensions. If $V_i(x,y) = 1$ (resp. $V_i(x,y) = -1$), then the comparison of x and y for criterion i leads (resp. does not lead) to a veto. Again, according to the developments in [1], if $g_i(x) + wv < g_i(y) \leq g_i(x) + v$ then it cannot be determined whether we have a veto situation between x and y or not, and $V_i(x,y) = 0$. Figure 1 represents the local concordance and veto indexes for a fixed $g_i(x)$.

To take into account these veto effects, the overall outranking index $\widetilde{S}''$ is defined as follows for all pairs of alternatives $(x,y) \in X \times X$:

$$\widetilde{S}''(x,y) = \min\big\{ \sum_{i \in N} w_i C_i'(x,y), -V_1(x,y), \ldots, -V_n(x,y) \big\}. \tag{10}$$

The min operator in Formula (10) tranlsates the conjunction between the overall concordance and the negated local veto indexes for each criterion. In the case of absence of veto on all the criteria ($V_i = -1\ \forall i \in N$), we have $\widetilde{S}''(x,y) = \widetilde{S}'(x,y)$.

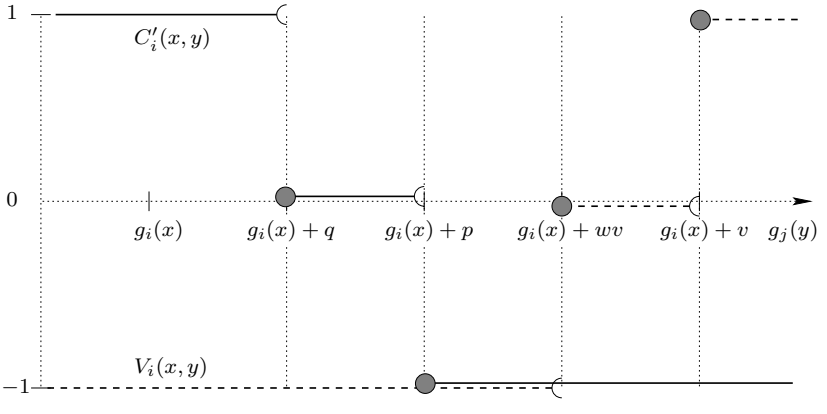


Fig. 1. Local concordance and veto indexes for a fixed $g_i(x)$

Similarly as in Section 3, the three values taken by the local veto function can be represented by means of two binary variables $\alpha'_i(x,y) \in \{0,1\}$ and $\beta'_i(x,y) \in \{0,1\}$ $(\forall i \in N, \forall x \neq y \in X)$ s.t.

$$V_i(x,y) = \alpha'_i(x,y) - \beta'_i(x,y).$$

Recalling that $wv < v$, conditions (9) can then be rewritten as follows as linear constraints $(\forall x \neq y \in X, \forall i \in N)$:

$$\begin{cases} (-1+wv)(1-\beta'_i(x,y)) < g_i(x) - g_i(y) + wv \leq (1+wv)\beta'_i(x,y); \\ (-1+v)\alpha'_i(x,y) < \ g_i(x) - g_i(y) + v \ \leq (1+v)(1-\alpha'_i(x,y)). \end{cases} \tag{11}$$

To represent Formula (10) as a set of linear constraints, we need to introduce some further binary variables $z_0(x,y)$ and $z_i(x,y)$ $(\forall x \neq y \in X, \forall i \in N)$ s.t.:

$$\widetilde{S}''(x,y) = \begin{cases} -V_k(x,y) & \text{if} \quad z_k(x,y)=1 \text{ and } z_i(x,y)=0 \ \ \forall i \in N \cup \{0\} \setminus \{k\}; \\ \sum\limits_{i \in N} w_i C'_i(x,y) & \text{if} \quad z_0(x,y)=1 \text{ and } z_i(x,y)=0 \ \ \forall i \in N. \end{cases}$$

This leads to the following linear constraints:

$$\begin{array}{ll} \widetilde{S}''(x,y) \leq \sum\limits_{i \in N} w_i C'_i(x,y) & \forall x \neq y \in X; \\ \widetilde{S}''(x,y) \leq -(\alpha'_i(x,y) - \beta'_i(x,y)) & \forall x \neq y \in X, \forall i \in N; \\ \sum\limits_{i \in N} w_i C'_i(x,y) \leq 2(1 - z_0(x,y)) + \widetilde{S}''(x,y) & \forall x \neq y \in X; \\ -(\alpha'_i(x,y) - \beta'_i(x,y)) \leq 2(1 - z_i(x,y)) + \widetilde{S}''(x,y) & \forall x \neq y \in X, \forall i \in N; \\ \sum\limits_{i=0}^{n} z_i(x,y) = 1 & \forall x \neq y \in X. \end{array} \tag{12}$$

Due to the last condition of Constraints (12), there exists a unique $k \in N \cup \{0\}$ s.t. $z_k = 1$ and $z_i = 0$ for $i \in N \cup \{0\} \setminus \{k\}$. Besides, if $\sum\limits_{i \in N} w_i C'_i(x,y) < -V_i(x,y)$ holds for all $i \in N$, then $z_i(x,y) = 0$ for all $i \in N$ and $z_0(x,y) = 1$ (which implies that $\widetilde{S}''(x,y) = \sum\limits_{i \in N} w_i C'_i(x,y)$). Furthermore, if $\exists k \in N \cup \{0\}$ s.t. $-V_k(x,y) < \sum\limits_{i \in N} w_i C'_i(x,y)$ and $-V_k(x,y) < -V_i(x,y)$ $(\forall i \in N \setminus \{k\})$, then $z_k(x,y) = 1$ (which implies that $\widetilde{S}''(x,y) = -V_k(x,y)$).

Constraints (12) only represent Formula (10) if all criteria have strictly positive weights. Note also that the first and the third condition of Constraints (12) can easily be linearised as in Section 3.

These considerations lead to the formulation of the mixed integer program **MIP3**, whose objective is to minimise a non-negative variable ε representing the maximal gap between $\widetilde{S}''(x,y)$ and $\sum\limits_{i \in N} w_i C'_i(x,y)$, for all $x \neq y \in X$ where the bipolar-valued outranking relation requires no veto. As $\widetilde{S}''(x,y)$ equals -1 or 0 in veto situations, no gap is considered on these values. Remember that all the weights are supposed to be strictly positive.

MIP3:

Variables:

$\varepsilon \geq 0$

$g_i(x) \in [0,1]$ $\quad \forall i \in N, \forall x \in X$

$w_i \in]0,1]$ $\quad \forall i \in N$

$\alpha_i(x,y) \in \{0,1\}$ $\quad \forall i \in N, \forall x \neq y \in X$

$\beta_i(x,y) \in \{0,1\}$ $\quad \forall i \in N, \forall x \neq y \in X$

$\alpha'_i(x,y) \in \{0,1\}$ $\quad \forall i \in N, \forall x \neq y \in X$

$\beta'_i(x,y) \in \{0,1\}$ $\quad \forall i \in N, \forall x \neq y \in X$

$w''_i(x,y) \in [-1,1]$ $\quad \forall i \in N, \forall x \neq y \in X$

$z_i(x,y) \in \{0,1\}$ $\quad \forall i \in N \cup \{0\}, \forall x \neq y \in X$

Parameters:

$q \in]0,p[$

$p \in]q,1[$

$wv \in]p,1[$

$v \in]wv,1[$

$\widetilde{S}''(x,y) \in [0,1]$ $\quad \forall x \neq y \in X$

$\delta \in]0,q[$

Objective function:

$\min \varepsilon$

Constraints:

s.t. $\sum_{i=1}^{n} w_i = 1$

$-w_i \leq w''_i(x,y)$ $\quad \forall x \neq y \in X, \forall i \in N$

$w''_i(x,y) \leq w_i$ $\quad \forall x \neq y \in X, \forall i \in N$

$w_i + \alpha_i(x,y) - \beta_i(x,y) - 1 \leq w''_i(x,y)$ $\quad \forall x \neq y \in X, \forall i \in N$

$w''_i(x,y) \leq -w_i + \alpha_i(x,y) - \beta_i(x,y) + 1$ $\quad \forall x \neq y \in X, \forall i \in N$

$-[\alpha_i(x,y) + \beta_i(x,y)] \leq w''_i(x,y)$ $\quad \forall x \neq y \in X, \forall i \in N$

$w''_i(x,y) \leq \alpha_i(x,y) + \beta_i(x,y)$ $\quad \forall x \neq y \in X, \forall i \in N$

$\widetilde{S}''(x,y) - \varepsilon \leq \sum_{i \in N} w''_i$ $\quad \forall x \neq y \in X$

$\widetilde{S}''(x,y) \leq -(\alpha'_i(x,y) - \beta'_i(x,y))$ $\quad \forall x \neq y \in X, \forall i \in N$

$\sum_{i \in N} w''_i \leq 2(1 - z_0(x,y)) + \widetilde{S}''(x,y) + \varepsilon$ $\quad \forall x \neq y \in X$

$-(\alpha'_i(x,y) - \beta'_i(x,y)) \leq 2(1 - z_i(x,y)) + \widetilde{S}''(x,y)$ $\quad \forall x \neq y \in X, \forall i \in N$

$\sum_{i=0}^{n} z_i(x,y) = 1$ $\quad \forall x \neq y \in X$

$(-1+q)(1 - \alpha_i(x,y)) + \delta \leq g_i(x) - g_i(y) + q$ $\quad \forall x \neq y \in X, \forall i \in N$ (c)

$g_i(x) - g_i(y) + q \leq (1+q)\alpha_i(x,y)$ $\quad \forall x \neq y \in X, \forall i \in N$

$(-1+p)\beta_i(x,y) + \delta \leq g_i(x) - g_i(y) + p$ $\quad \forall x \neq y \in X, \forall i \in N$

$g_i(x) - g_i(y) + p \leq (1+p)(1 - \beta_i(x,y))$ $\quad \forall x \neq y \in X, \forall i \in N$

$(-1+wv)(1 - \beta'_i(x,y)) + \delta \leq g_i(x) - g_i(y) + wv$ $\quad \forall x \neq y \in X, \forall i \in N$

$g_i(x) - g_i(y) + wv \leq (1+wv)\beta'_i(x,y)$ $\quad \forall x \neq y \in X, \forall i \in N$

$(-1+v)\alpha'_i(x,y) + \delta \leq g_i(x) - g_i(y) + v$ $\quad \forall x \neq y \in X, \forall i \in N$

$g_i(x) - g_i(y) + v \leq (1+v)(1 - \alpha'_i(x,y))$ $\quad \forall x \neq y \in X, \forall i \in N$

4.1 Example

Let us consider the bipolar-valued outranking relation $\widetilde{S}$ on $X = \{a, b, c\}$ of Table 1 and fix $q = 0.1$, $p = 0.2$, $wv = 0.6$ and $v = 0.8$. Let us first try to represent $\widetilde{S}$ by model $\mathcal{M}_2$. For $n = 4$, the value of the objective function for the optimal solution of **MIP2** equals 0.593. The weights w_3 and w_4 equal 0. Table 2 summarises the outranking relation associated with its optimal solution determined by solving **MIP2** for $n = 4$. One can easily check that $\widetilde{S}$ and $\widetilde{S}^*$ differ by at most 0.593. This shows that this outranking relation is not representable by $\mathcal{M}_2$ and at most 4 criteria. We therefore switch to the more general model $\mathcal{M}_3$ with two preference and two veto thresholds.

For $n = 4$ the value of the objective function for the optimal solution of **MIP3** equals 0. This means that $\widetilde{S}$ can be built from a performance table with 4 criteria via $\mathcal{M}_3$, given the above thresholds. For lower values of n, the objective function for the optimal solution is strictly positive. Table 3 shows the performances of the three alternatives and the weights which allow to construct $\widetilde{S}$ via model $\mathcal{M}_3$. A veto situation occurs between a and c on criterion g_4 ($\widetilde{S}(c, a) = -1$).

Table 1. Given $\widetilde{S}$

$\widetilde{S}$	a	b	c
a	·	0.258	-0.186
b	0.334	·	0.556
c	-1.000	0.036	·

Table 2. Approximative outranking relation $\widetilde{S}^*$ via **MIP1bis** for $n = 4$

$\widetilde{S}^*$	a	b	c	g_1	g_2
a	·	0.407	0.407	0.280	0.000
b	0.296	·	1.000	0.090	1.000
c	-0.407	0.407	·	0.000	0.200
w_i				0.704	0.296

Table 3. Performances and weights to construct $\widetilde{S}$ via model $\mathcal{M}_3$

	g_1	g_2	g_3	g_4
a	0.000	0.000	0.000	1.000
b	0.400	0.100	0.090	0.590
c	0.200	0.290	0.000	0.000
w_i	0.149	0.444	0.074	0.333

References

1. Bisdorff, R., Meyer, P., Roubens, M.: Rubis: a bipolar-valued outranking method for the best choice decision problem. 4OR, Quaterly Journal of the Belgian, French and Italian Operations Research Societies 6(2) (June 2008), doi:10.1007/s10288-007-0045-5
2. Roy, B.: Méthodologie multicritère d'aide à la décision. Ed. Economica, collection Gestion (1985)

A Performance Study of Task Scheduling Heuristics in HC Environment

Ehsan Ullah Munir, Jianzhong Li, Shengfei Shi,
Zhaonian Zou, and Qaisar Rasool

School of Computer Science and Technology, Harbin Institute of Technology,
Harbin 150001, China
ehsanmunnir@gmail.com, lijzh@hit.edu.cn, shengfei@hit.ed.cn,
zouzhaonian@gmail.com, qrasool@yahoo.com

Abstract. Heterogeneous computing (HC) environment consists of different resources connected with high-speed links to provide a variety of computational capabilities for computing-intensive applications having multifarious computational requirements. The problem of optimal assignment of tasks to machines in HC environment is proven to be NP-complete requiring use of heuristics to find the near optimal solution. In this work we conduct a performance study of task scheduling heuristics in HC environment. Overall we have implemented 16 heuristics, among them 7 are proposed in this paper. Based on experimental results we specify the circumstances under which one heuristic will outperform the others.

Keywords: Heterogeneous computing, Task scheduling, Performance evaluation, Task Partitioning heuristic.

1 Introduction

Heterogeneous computing (HC) environment consists of different resources connected with high-speed links to provide a variety of computational capabilities for computing-intensive applications having multifarious computational requirements [1]. In HC environment an application is decomposed into various tasks and each task is assigned to one of the machines, which is best suited for its execution to minimize the total execution time. Therefore, an efficient assignment scheme responsible for allocating the application tasks to the machines is needed; formally this problem is named task scheduling [2]. Developing such strategies is an important area of research and it has gained a lot of interest from researchers [3, 4]. The problem of task scheduling has gained tremendous attention and has been extensively studied in other areas such as computational grids [5] and parallel programs [6].

The problem of an optimal assignment of tasks to machines is proven to be NP-complete requiring use of heuristics to find the near-optimal solution [7]. Plethora of heuristics has been proposed for assignment of tasks to machines in HC environment [7, 8, 9]. Each heuristic has different underlying assumptions to

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 214–223, 2008.

produce near optimal solution however no work reports which heuristic should be used for a given set of tasks to be executed on different machines.

Provided with a set of tasks $\{t_1, t_2, ..., t_m\}$, a set of machines $\{m_1, m_2, ..., m_n\}$ and expected time to compute of each task t on each machine m_j, $ETC\,(t_i, m_j)$, $(1 \leq i \leq m, 1 \leq j \leq n)$, in the current study we find out the task assignment strategy that gives the minimum makespan.

For task selection in heterogeneous environment different criteria can be used, e.g. minimum, maximum or average of expected execution time across all machines. In current work we propose a new heuristic based on task partitioning, which consider minimum (min), maximum (max), average (avg), median (med) and standard deviation (std) of expected execution time of task on different machines as selection criteria. We call each selection criterion a key. Each heuristic uses only one key. Scheduling process for the proposed heuristics works like this; all the tasks are sorted in decreasing order of their key, then these tasks are partitioned into k segments and after this scheduling is performed in each segment.

A large number of experiments were conducted on synthetic datasets; Coefficient of Variation (COV) based method was used for generating synthetic datasets, which provides greater control over spread of heterogeneity [10]. A comparison among existing heuristics is conducted and new heuristics are proposed. Extensive simulation results illustrate the circumstances when one heuristic would outperform other heuristics in terms of average makespan. This work is intended to establish basis for selecting a heuristic for any given ETC.

2 Related Work

Many heuristics have been developed for task scheduling in heterogeneous computing environments. Min-min [11] gives the highest priority to the task for scheduling, which can be completed earliest. The idea behind Min-min heuristic is to finish each task as early as possible and hence, it schedules the tasks with the selection criterion of minimum earliest completion time. Max-min [11] heuristic is very similar to the Min-min, which gives the highest priority to the task with the maximum earliest completion time for scheduling. The idea behind Max-min is that overlapping long running tasks with short-running ones. The Heaviest Task First (HTF) heuristic [12] computes each tasks minimum execution time on all machines and the task with the maximum execution time is selected. The selected task is the heaviest task among all tasks (note that the Max-Min algorithm selects the task with the latest minimum completion time, which may not be the heaviest one). Then this heaviest task is assigned to the machine on which this task has minimum completion time. Eight dynamic mapping heuristics are given and compared in [3], however the problem domain considered there involves priorities and multiple deadlines. In Segmented Min-min heuristic [8] the tasks are divided into four groups based on their minimum, maximum or average expected execution time, and then Min-min is applied on each group for scheduling. In [1] the comparison of eleven heuristics is given and the Min-min

heuristic is declared the best among all the other heuristics considered based on makespan criterion. Minimum standard deviation first heuristics is proposed in [13] where the task having the minimum standard deviation is scheduled first.

Current research is different from the related work as different keys are used here as a selection criterion suiting the ETC type. Moreover for any given ETC we provide the near optimal solution by using the heuristic best suited for a specific ETC type.

3 Problem Definition

Let $T = \{t_1, t_2, ..., t_m\}$ be a set of tasks, $M = \{m_1, m_2, ..., m_n\}$ be a set of machines, and the expected time to compute (ETC) is a $m \times n$ matrix where the element ETC_{ij} represents the expected execution time of task t_i on machine m_j. For clarity, we denote ETC_{ij} by $ETC\,(t_i, m_j)$ in the rest of the paper. Machine availability time, $MAT\,(m_j)$, is the earliest time machine m_j can complete the execution of all the tasks that have previously been assigned to it (based on the ETC entries for those tasks). The completion time (CT) of task t_i on machine is equal to the execution time of t_i on plus the machine availability time of m_j i.e.

$$CT\,(t_i, m_j) = ETC\,(t_i, m_j) + MAT\,(m_j)$$

Makespan (MS) is equal to the maximum value of the completion time of all tasks i.e.

$$MS = \max MAT\,(m_j) \text{ for } (1 \leq j \leq n)$$

Provided with T, M and ETC our objective is to find the task assignment strategy that minimizes makespan.

4 Task Partitioning Heuristic

In heterogeneous environment for task selection different criteria can be used, examples are minimum, maximum or average of expected execution time across all machines. In task partitioning heuristic we use minimum (min), maximum (max), average (avg), median (med) and standard deviation (std) of expected execution time of task on different machines as selection criteria; hereafter referred to as key. Given a set of tasks $T = \{t_1, t_2, ..., t_m\}$, a set of machines $M = \{m_1, m_2, ..., m_n\}$, expected time to compute (ETC) matrix then the working of proposed heuristic can be explained as follows: we compute the sorting key for each task (for each heuristic only one key will be used for sorting), then we sort the tasks in decreasing order of their sorting key. Next the tasks are partitioned into k disjoint equal sized groups. At the end, tasks are scheduled in each group g_x using the following procedure:

Procedure1

a) for each task t_i in a group g_x find machine m_j which completes the task at earliest.
b) machine m_j is available i.e. no task is assigned to machine then assign task to machine and remove it from list of tasks.
c) If there is already task t_k assigned to machine m_j i.e. machine is not available then compute the difference between the minimum earliest completion time and the second smallest earliest completion time on all machines for t_i and t_k respectively.
1) If the difference value for t_i is larger than that of t_k then t_i is assigned to machine m_j.
2) If the difference value for t_i is less than that of t_k, then no changes to the assignment.
3) If the differences are equal, we compute the difference between the minimum earliest completion time and the third smallest earliest completion time for t_i and t_k respectively. And repeat 1-3. Every time if step 3 is selected, the difference between the minimum earliest completion time and the next earliest completion time (e.g. the fourth, the fifth) for t_i and t_k are computed respectively. If all the differences are the same then the task is selected deterministically i.e. the oldest task is chosen.

Now the proposed Task partitioning algorithm can be summed up in the following steps:

Task Partitioning Heuristic

1) Compute the sorting key for each task:
Sub-policy1 (avg): Compute the average value of each row in ETC matrix
$key_i = \sum_j ETC\,(t_i, m_j)\,/n.$
Sub-policy2 (min): Compute the minimum value of each row in ETC matrix
$key_i = \min_j ETC\,(t_i, m_j)\,.$
Sub-policy3 (max): Compute the maximum value of each row in ETC matrix
$key_i = \max_j ETC\,(t_i, m_j)\,.$
Sub-policy4 (med): Compute the median value of each row in ETC matrix
$key_i = \underset{j}{med}\, ETC\,(t_i, m_j)\,.$
Sub-policy5 (std): Compute the standard deviation value of each row in ETC matrix
$key_i = \underset{j}{std}\, ETC\,(t_i, m_j)\,.$
2) Sort the tasks in decreasing order of their sorting key (for each heuristic only one key will be used for sorting). 3) Partition the tasks evenly into k segments.
4) Apply the Procedure1 for scheduling each segment.

Table 1. Summary of compared heuristics

No	Name	Reference	No	Name	Reference
H1	TPAvg	New	H9	Smm-avg	[8]
H2	TPMin	New	H10	Smm-min	[8]
H3	TPMax	New	H11	Smm-max	[8]
H4	TPMed	New	H12	Smm-med	New
H5	TPStd	New	H13	Smm-std	New
H6	Min-min	[11]	H14	MCT	[7]
H7	Max-min	[11]	H15	minSD	[13]
H8	Sufferage	[7]	H16	HTF	[12]

4.1 Heuristics Notation

In task partitioning heuristic tasks are sorted based on average, minimum, maximum, median and standard deviation, and each heuristic is named as TPAvg, TPMin, TPMax, TPMed and TPStd. The algorithms Segmented min-min (med) and Segmented min-min (std) are also implemented for the evaluation purpose. The naming conventions and source information for all existing and proposed heuristics are detailed in Table 1.

5 Experimental Results and Analysis

5.1 Dataset

In the experiments, COV based ETC generation method is used to simulate different HC environments by changing the parameters μ_{task}, V_{task} and $V_{machine}$, which represent the mean task execution time, the task heterogeneity, and the machine heterogeneity, respectively. The COV based method provides greater control over the spread of the execution time values than the common range-based method used previously [1].

The COV-based ETC generation method works as follows [10]: First, a task vector, q, of expected execution times with the desired task heterogeneity is generated following gamma distribution with mean μ_{task} and standard deviation $\mu_{task} * V_{task}$. The input parameter is used to set the average of the values in q. The input parameter μ_{task} is the desired coefficient of variation of the values in q. The value of V_{task} quantifies task heterogeneity, and is larger for high task heterogeneity. Each element of the task vector q is then used to produce one row of the ETC matrix following gamma distribution with mean $q[i]$ and standard deviation $q[i] * V_{machine}$ such that the desired coefficient of variation of values in each row is $V_{machine}$, another input parameter. The value of $V_{machine}$ quantifies machine heterogeneity, and is larger for high machine heterogeneity.

5.2 Comparative Performance Evaluation

The performance of the heuristic algorithm is evaluated by the average makespan of 1000 results on 1000 ETCs generated by the same parameters. In all the experiments, the size of ETCs is 512×16, the value of $k = 3$ (i.e. tasks are partitioned into 3 segments) the mean of task execution time μ_{task} is 1000, and the task COV V_{task} is in $[0.1, 2]$ while the machine COV $V_{machine}$ is in $[0.1, 1.1]$.

The motivation behind choosing such heterogeneous ranges is that in real situation there is more variability across execution times for different tasks on a given machine than the execution time for a single task across different machines. The range bar for the average makespan of each heuristic shows a 95% confidence interval for the corresponding average makespan. This interval represents the likelihood that makespans of task assignment for that type of heuristic fall within the specified range. That is, if another ETC matrix (of the same type) is generated, and the specified heuristic generates a task assignment, then the

makespan of the task assignment would be within the given interval with 95% certainty. In our experiments we have also considered two metrics in comparison of heuristics. Such metrics have also been considered by [9]

- The number of best solutions (denoted by NB) is the number of times a particular method was the only one that produced the shortest makespan.
- The number of best solutions equal with another method (denoted by NEB), which counts those cases where a particular method produced the shortest makespan but at least one other method also achieved the same makespan. NEB is the complement to NB.

The proposed heuristics are compared with 11 existing heuristics. Experiments are performed with different ranges of task and machine heterogeneity. In the

Table 2. NB and NEB values table when fix $V_{task} = 2$

COV of machines		H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16
0.1	NB	86	197	169	78	**245**	0	0	96	0	0	0	0	0	0	0	4
	NEB	97	27	48	92	29	0	2	18	0	0	0	0	0	0	0	2
0.3	NB	101	**252**	112	132	90	0	0	213	0	1	0	0	0	0	0	0
	NEB	62	54	48	62	52	0	1	49	0	0	0	0	0	0	0	4
0.5	NB	101	**352**	98	106	65	0	0	92	0	1	1	1	1	0	0	19
	NEB	105	84	104	103	99	0	1	90	1	0	1	1	0	0	0	10
0.7	NB	82	**350**	62	89	47	0	0	45	1	2	4	1	2	0	0	146
	NEB	100	59	98	96	99	0	2	89	0	0	2	1	1	0	0	32
0.9	NB	60	199	43	62	44	0	0	11	5	2	2	4	0	0	0	**381**
	NEB	103	78	115	103	110	0	14	94	1	0	2	0	1	2	0	90
1.1	NB	17	69	22	21	16	0	0	9	0	1	0	3	1	0	0	**575**
	NEB	167	156	160	163	160	0	47	156	1	0	3	1	2	5	0	202

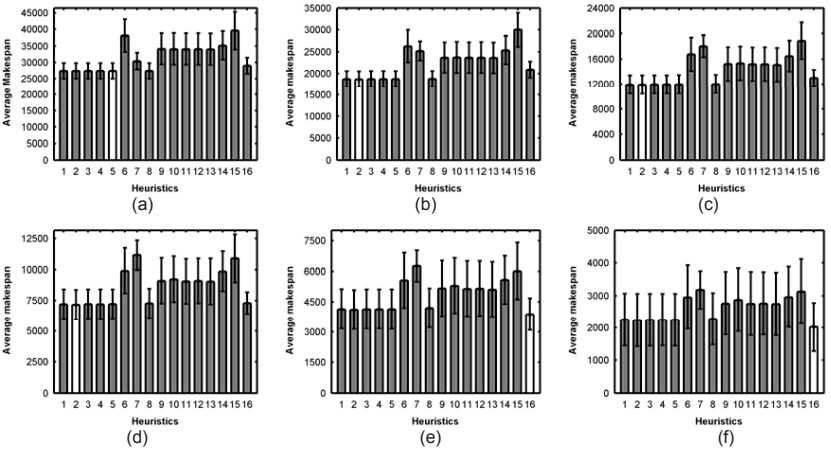


Fig. 1. Average makespan of the heuristics when $V_{task} = 2$ and $V_{machine}$ = (a) 0.1; (b) 0.3; (c) 0.5; (d) 0.7; (e) 0.9; (f) 1.1

first experiment we have fixed the value of $V_{task} = 2$ while increasing the value of $V_{machine}$ from 0.1 to 1.1 with a step size of 0.2. The results of NB and NEB are shown in the Table 2 (best values shown in bold). From the values we can

see that for high values of $V_{machine}$ H16 performs better. And in all other cases one of the proposed heuristic H2 or H5 outperforms all other heuristics. Fig. 1 gives the comparison of average makespan of the all heuristics considered.

Secondly, we have fixed the value of $V_{task} = 1.1$ and increased the value of $V_{machine}$ from 0.1 to 1.1 with increment of 0.2 in each step. The results of NB and NEB are shown in the Table 3 which confirm that in all the cases one of the proposed heuristic H2 or H5 is best. Fig. 2 gives the comparison of average makespan of all the heuristics considered.

Table 3. NB and NEB values table when fix $V_{task} = 1.1$

COV of machines		H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16
0.1	NB	141	159	150	150	**372**	0	0	0	0	0	0	0	0	0	0	0
	NEB	24	2	5	21	6	0	0	0	0	0	0	0	0	0	0	0
0.3	NB	139	**284**	199	161	211	0	0	0	0	0	0	0	0	0	0	0
	NEB	2	4	2	3	1	0	0	0	0	0	0	0	0	0	0	0
0.5	NB	129	**445**	154	127	142	0	0	0	0	0	0	0	0	0	0	0
	NEB	1	2	2	0	1	0	0	0	0	0	0	0	0	0	0	0
0.7	NB	84	**613**	97	82	102	0	0	0	3	10	1	2	0	0	0	0
	NEB	3	2	4	3	1	0	0	0	0	0	0	0	0	0	0	0
0.9	NB	78	**586**	80	63	91	0	0	0	8	59	5	14	1	0	0	2
	NEB	6	8	6	7	4	0	0	1	0	2	0	0	0	0	0	1
1.1	NB	66	**505**	76	73	63	0	0	1	28	24	4	24	4	0	0	92
	NEB	20	24	17	16	14	0	0	10	3	0	1	1	1	0	0	11

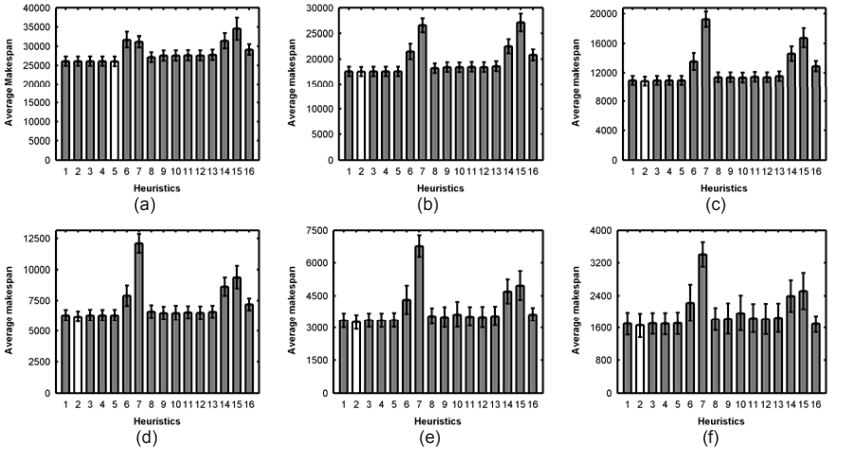


Fig. 2. Average makespan of the heuristics when $V_{task} = 1.1$ and $V_{machine}$ = (a) 0.1; (b) 0.3; (c) 0.5; (d) 0.7; (e) 0.9; (f) 1.1

In the third experiment V_{task} is fixed to 0.6 and value of $V_{machine}$ is increased from 0.1 to 1.1 with step size of 0.2. As shown in the Table 4, proposed heuristic H5 outperforms all other heuristics in every case. Fig. 3 gives the comparison of average makespan of all the heuristics. The results for fixing $V_{task} = 0.1$ are same with $V_{task} = 0.6$ and hence not shown due to space limitations. From these experiments we conclude that in most of circumstances one of the proposed heuristics

Table 4. NB and NEB values table when fix $V_{task} = 0.6$

COV of machines		H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16
0.1	NB	81	80	78	79	**682**	0	0	0	0	0	0	0	0	0	0	0
	NEB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.3	NB	73	42	143	76	**663**	0	0	0	0	0	0	0	0	0	0	0
	NEB	1	1	3	0	1	0	0	0	0	0	0	0	0	0	0	0
0.5	NB	84	20	254	118	**520**	0	0	0	0	0	0	0	0	0	0	0
	NEB	3	0	2	0	3	0	0	0	0	0	0	0	0	0	0	0
0.7	NB	127	13	285	130	**441**	0	0	0	0	0	0	0	0	0	0	0
	NEB	2	0	3	1	2	0	0	0	0	0	0	0	0	0	0	0
0.9	NB	150	33	313	144	**354**	0	0	0	0	0	0	0	0	0	0	0
	NEB	2	0	2	4	4	0	0	0	0	0	0	0	0	0	0	0
1.1	NB	138	124	245	158	**313**	0	0	0	0	6	0	0	0	0	0	1
	NEB	4	9	5	8	5	0	0	0	0	1	0	0	0	0	0	0

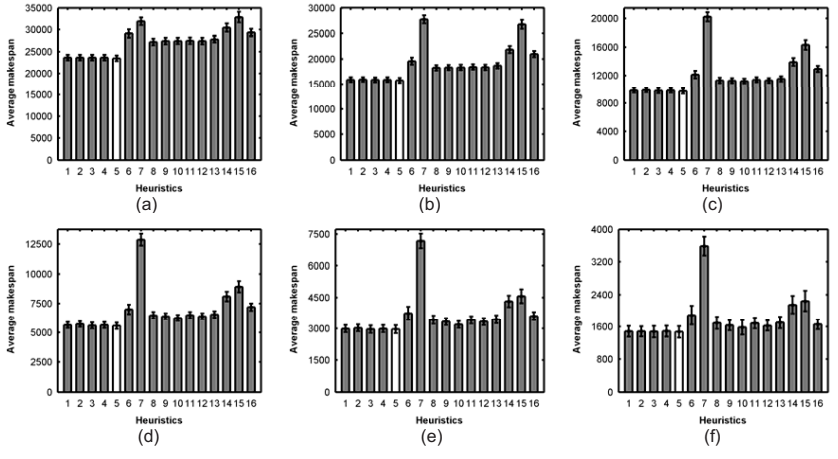


Fig. 3. Average makespan of the heuristics when $V_{task} = 0.6$ and $V_{machine}$ = (a) 0.1; (b) 0.3; (c) 0.5; (d) 0.7; (e) 0.9; (f) 1.1

H2 or H5 outperforms the existing heuristics in terms of average makespan. In the remaining cases H16 performs better.

5.3 Algorithm to Find Best Heuristic

Based on the values of V_{task} and $V_{machine}$ we divide ETC into three different regions. If the values of V_{task} and are high (here $V_{tast} = 2$ and $0.9 \leq V_{machine} \leq 1.1$) then ETC falls in the region 1, if either of them is medium (here $V_{mask} = 1.1$ or $0.3 \leq V_{machine} \leq 0.7$) then it falls in region 2 and if either of them is low (here $0.1 \leq V_{machine} \leq 0.6$ or $0.1 \leq V_{machine} \leq 0.2$) then it falls in region 3. Fig. 4 shows the three regions and best heuristic for each region.

The procedure for finding a best heuristic is given below in algorithm Best Heuristic, which suggests the best heuristic depending on ETC type.

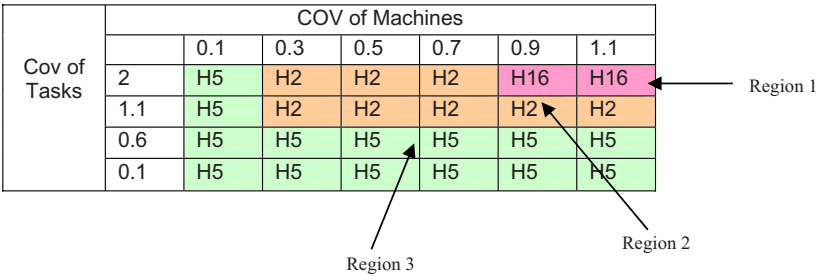


Fig. 4. Division of ETC in different regions

Best Heuristic

Input: expected time to compute matrix (ETC)
Output: best heuristic
Compute the V_{mask} and $V_{machine}$
if V_{mask} is high and $V_{machine}$ is high then
ETC belongs to region1
if V_{mask} is medium or $V_{machine}$ is medium then
ETC belongs to region2
if V_{mask} is low or $V_{machine}$ is low then
ETC belongs to region3
end if
switch(region)
case region1: return H16
case region2: return H2
case region3: return H5
end switch

6 Conclusions

Optimal assignment of tasks to machines in a HC environment has been proven to be a NP-complete problem. It requires the use of efficient heuristics to find near optimal solutions. In this paper, we have proposed, analyzed and implemented seven new heuristics. A comparison of the proposed heuristics with the existing heuristics was also performed in order to identify the circumstances in which one heuristic outperforms the others. The experimental results demonstrate that in most of the circumstances one of the proposed heuristics H2 or H5 outperforms all the existing heuristics. Based on these experimental results, we are also able to suggest, given an ETC, which heuristic should be used to achieve the minimum makespan.

Acknowledgments

This work is supported by the Key Program of The National Natural Science Foundation of China under Grant No.60533110 and 60703012; The National Grand Fundamental Research 973 Program of China, Grant No.2006CB303000; The Heilongjiang Province Scientific and Technological Special Fund for Young Scholars, Grant No.QC06C033. COMSATS Institute of Information Technology, Pakistan (CIIT) provides PhD scholarship for Mr. Ehsan Ullah Munir.

References

1. Braun, T.D., et al.: A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. Journal of Parallel and Distributed Computing 61, 810–837 (2001)
2. El-Rewini, H., Lewis, T.G., Ali, H.H.: Task Scheduling in Parallel and Distributed Systems. PTR Prentice Hall, New Jersey (1994)
3. Kim, J.K., et al.: Dynamically Mapping Tasks with Priorities and Multiple Deadlines in a Heterogeneous Environment. Journal of Parallel and Distributed Computing 67, 154–169 (2007)
4. Kwok, Y.K., et al.: A semi-static approach to mapping dynamic iterative tasks onto heterogeneous computing system. Journal of Parallel and Distributed Computing 66, 77–98 (2006)
5. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufman Publishers, San Francisco (1999)
6. Kwok, Y.K., Ahmad, I.: Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors. ACM Computing Surveys 31, 406–471 (1999)
7. Maheswaran, M., et al.: Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogeneous Computing Systems. In: Proceedings of the 8th IEEE Heterogeneous Computing Workshop, pp. 30–44 (1999)
8. Wu, M.Y., Shu, W., Zhnag, H.: Segmented min-min: A Static Mapping Algorithm for Meta-Tasks on Heterogeneous Computing Systems. In: Proceedings of the 9th Heterogeneous Computing Workshop, pp. 375–385 (2000)
9. Sakellariou, R., Zhao, H.: A Hybrid Heuristic for Dag Scheduling on Heterogeneous Systems. In: Proceedings of the 13th Heterogeneous Computing Workshop (2004)
10. Ali, S., et al.: Task Execution Time Modeling for Heterogeneous Computing Systems. In: Proceedings of the 9th Heterogeneous Computing Workshop, pp. 185–200 (2000)
11. Freund, R.F., et al.: Scheduling Resources in Multi-User, Heterogeneous, Computing Environments with Smartnet. In: Proceedings of the 7th Heterogeneous Computing Workshop, pp. 184–199 (1998)
12. Yarmolenko, V., Duato, J., Panda, D.K., Sadayappan, P.: Characterization and Enhancement of Static Mapping Heuristics for Heterogeneous Systems. In: International Conference on Parallel Processing, pp. 437–444 (2000)
13. Luo, P., Lu, K., Shi, Z.Z.: A Revisit of Fast Greedy Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Computing Systems. Journal of Parallel and Distributed Computing 67, 695–714 (2007)

Performance Evaluation in a Queueing System $M_2/G/1$

Naima Hamadouche[1] and Djamil Aissani[2]

Laboratory of Modelisation and Optimization of Systems (LAMOS)
University of Bejaia 06000, Algeria
naima_maths@yahoo.fr, lamos_bejaia@hotmail.com

Abstract. In this communication, we use the strong stability method to approximate the characteristics of the $M_2/G/1$ queue with preemptive resume priority by those of the $M/G/1$ one. For this, we first prove the stability fact and next obtain quantitative stability estimates with an exact computation of constants.

Keywords: Strong stability, Approximation, Preemptive priority, Markov chain.

1 Introduction

Queueing phenomena occur in several real situations when resources can not immediately render the current or the kind of service required by their users. The theory of queues is particulary well adapted to the study of the performance of computer systems and communication networks. Such systems often trait classes of request of different priorities. Nowadays, the introduction of priorities in such systems is frequent and often motivated by the need to increase their performance and quality of service. A complete presentation of the area was the subject of monographs of N.K.Jaiswal [7] or of B.V.Gneedenko and al [5]. Nevertheless, these non markovien systems are complex then difficult to study and their characteristic are obtained in a very complicated way of the parameters of the system [7].

In addition, the including some complex systems in queueing networks, dos not allow the use of the existing queueing networks models (with product form solutions). This why it is interesting to study the proximity of characteristics of some complex systems by those of the simpler and more exploitable one. The purpose of this paper is to obtain the conditions and estimations of strong stability of an imbedded Markov chain in an $M_2/G/1$ system with a preemptive priority. This is to approximate this system by the $M/G/1$ model. Indeed, the characteristic of the queue $M/G/1$ are obtained in an explicit form and this last allows the use of the product form solutions.

In the stability theory, we establish the domain within a model may be used as a good approximation or idealization to the real system under consideration. The stability methods allow to investigate qualitative proprieties of the

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 224–233, 2008.

system, in particular its robustness. In addition, using these approach, bounds can be obtained in an explicit form and approximations can be made rigourously [13]. Indeed, measure of robustness of the system also need to be evaluated in comparison to the often studied measures of performance and efficiency [11]. The first results on the stability have been obtained by Rosseberg [12], Gnedenko [5], Franken [4] and Kennedy [10]. Afterwards, several papers have considered various situations and various approaches. Stoyen proposed the weak convergence method [13] used to investigate proprieties of stability of homogeneous Markov processus. Kalashinkov and Tsitsiashvili proposed the method of test functions [8] which consist in constructing a test function allowing to compare the behavior of the perturbed system (real model) with the non-perturbed system (ideal model). Borovkov proposed the renewal method [3] whose advantage comes from the fact that it allows to obtain theorems of ergodicity and stability with minimal conditions. Zolotariev and Rachev proposed the metric method [14] , [11]. Ipsen and Meyer proposed the uniform stability method [6] whose aim to analyze the sensitivity of individual stationary probabilities to perturbations in the transition probabilities of finite irreductible Markov chains. Kartashov and Aïssani proposed the strong stability method [1]. In contrast to other methods, this technique suppose that the perturbations of the transition kernel (associated to the Markov chain describing the system) is small with respect to a certain norm. Such strict conditions allows us to obtain better estimations on the characteristics of the perturbed chain. This article is strutted as follow. In the section 2, we clarify the Markov chains and their transition operation describing the system and the basic theorems of the strong stability method. In the section 3, we prove the strong stability in an $M/G/1$ queue. On other words we clarify the condition under which the M/G/1 system can be approximate the $M_2/G/1$ system with priority. The section 4 gives the error of approximation on the stationary distribution of the number of request when the intensity of the flux is sufficing small.

2 Description of the Systems

Let us consider a queueing system $M_2/G/1$ with preemptive priority. Priority and non priority request arrive at service mechanism in poisson streams with mean rates λ_1 and λ_2 respectively. The service of priority and non-priority request are distributed with probability density $b(t)$. The service of a non-priority request may be interrupted by the arrival of a priority request. When the later completes its service, the interrupted begin again its service if no priority request are waiting. The service time of the non priority request up to the interruption is distributed with probability density $b^*(t)$. The state of the $M_2/G/1$ queueing system with preemptive priority at time t can be described by using the method of imbedded Markov chain, for this we define:

X_{n+1}^i: the number of priority request(respectively non-priority request) in the system at instant t_{n+1}.
If $X_n^1 \neq 0 : t_{n+1}$ is the instant of "end of service" of priority request.

If $X_n^1 = 0$: t_{n+1} is the instant of end of service of non-priority request" or "instant of interruption of priority request"

$A_{n+1}^i,\ i = 1, 2$: is a random variable that represents the number of the priority (respectively non priority) request arriving during the $(n+1)^{th}$ service.

- If $t_{n+1}$is the instant of the end of service of priority (respectively non priority) request, the distribution of $A_{n+1}^i,\ i = 1, 2$ is:

$a_k^i = P(A_{n+1}^i = k) = \int_0^\infty \frac{(\lambda_i t)^k}{k!} e^{-\lambda_i t} b(t)dt.\ i = 1, 2$

-If t_{n+1} is the instant of interruption, the distribution of A_{n+1}^1 is:

$a_k^1 = P(A_{n+1}^1 = 1) = \int_0^\infty \lambda_1 t\ e^{-\lambda_1 t} b^*(t)dt.$

The random variables A_{n+1}^1, A_{n+1}^2 are independent between them.

Lemma 1. *The sequence* $(X_{n+1}^1,\ X_{n+1}^2)$ *forms a Markov chain of transition operator* $P_{k,l}(i,j)_{i,j\geq 0}$ *defined by:*

$$P_{k,l}(i,j) = \begin{cases} \bullet \int_0^\infty \frac{(\lambda_1 t)^{i-k+1}}{(i-k+1)!} \frac{(\lambda_2 t)^{j-l}}{(j-l)!} e^{-(\lambda_1+\lambda_2)t} b(t)dt, \\ if\ k > 0,\ j \geq l,\ l \geq 0,\ i \geq k-1. \\ \bullet \int_0^\infty e^{-\lambda_1 t} b(t)dt \int_0^\infty \frac{(\lambda_2 t)^{j-l+1}}{(j-l+1)!} e^{-\lambda_2 t} b(t)dt \\ + \int_0^\infty (\lambda_1 t) e^{-\lambda_1 t} b^*(t)dt \int_0^\infty \frac{(\lambda_2 t)^{j-l}}{(j-l)!} e^{-\lambda_2 t} b^*(t)dt. \\ if\ j \geq l-1,\ k = 0, i = 0,\ l \neq 0. \\ \bullet \frac{\lambda_1}{\lambda_1+\lambda_2} \int_0^\infty \frac{(\lambda_1 t)^i}{i!} \frac{(\lambda_2 t)^j}{j!} e^{-(\lambda_1+\lambda_2)t} b(t)dt + \frac{\lambda_2}{\lambda_1+\lambda_2} \times \\ \times [\int_0^\infty e^{-\lambda_1 t} b(t)dt \int_0^\infty \frac{(\lambda_2 t)^j}{j!} e^{-\lambda_2 t} b(t)dt \\ + \int_0^\infty (\lambda_1 t) e^{-\lambda_1 t} b^*(t)dt \int_0^\infty \frac{(\lambda_2 t)^j}{j!} e^{-\lambda_2 t} b^*(t)]dt, \\ if\ \ i \geq 0,\ j \geq 0,\ k = 0,\ l = 0. \end{cases}$$

Proof. In order to calculate $P_{k,l}(i,j)_{i,j}$ we consider the different cases:

Case 1 : $\mathbf{X_n^1 \neq 0},\ \mathbf{X_n^2 \geq 0}$

In this case t_{n+1} is the end of service of priority request, and the sequence $(X_{n+1}^1,\ X_{n+1}^2)$ is given by : $\begin{cases} X_{n+1}^1 = X_n^1 + A_{n+1}^1 - 1, \\ X_{n+1}^2 = X_n^2 + A_{n+1}^2. \end{cases}$

The probability of transition $P_{k,l}(i,\ j)$ is:

$$P_{k,l}(i,j) = P(A_{n+1}^1 = i-k+1, A_{n+1}^2 = j-l) = \int_0^\infty \frac{(\lambda_1 t)^{i-k+1}\ (\lambda_2 t)^{j-l}}{(i-k+1)!\,(j-l)!} e^{-(\lambda_1+\lambda_2)t} b(t)dt$$

Case 2 : $\mathbf{X_n^1 = 0},\ \mathbf{X_n^2 \neq 0}$

If t_{n+1} is the "end of service of non-priority request", the sequence (X_{n+1}^1, X_{n+1}^2) is given by : $\begin{cases} X_{n+1}^1 = 0, \\ X_{n+1}^2 = X_n^2 + A_{n+1}^2 - 1. \end{cases}$

That explains A :" no priority request arrives during the service of the non-priority request". The probability of A is : $P(A) = \int_0^\infty e^{-\lambda_1 t} b(t)dt$.

If t_{n+1} is the " instant of interruption of priority request", the sequence $(X_{n+1}^1,\ X_{n+1}^2)$ is given by: $\begin{cases} X_{n+1}^1 = 1, \\ X_{n+1}^2 = X_n^2 + A_{n+1}^2. \end{cases}$

That explains B : "priority request arrives during the service of non-priority request". The probability of B is : $P(B) = \int_0^\infty (\lambda_1 t)e^{-\lambda_1 t}b^*(t)dt$. Therefore the probability of transition $P_{k,l}(i,\ j)$is given by:

$$\begin{aligned}P_{k,l}(i,j) &= P(\mathbf{A})P(X_{n+1}^1 = 0,\ X_{n+1}^2 = X_n^2 + A_{n+1}^2 - 1 = j/X_n^1 = 0,\ X_n^2 = l)\\ &+ P(B)P(X_{n+1}^1 = X_n^1 + A_{n+1}^1 = 1,\ X_{n+1}^2 = X_n^2 + A_{n+1}^2 = j/X_n^1 = 1,\ X_n^2 = l)\\ &= \int_0^\infty e^{-\lambda_1 t}b(t)dt \int_0^\infty \frac{(\lambda_2 t)^{j-l+1}}{(j-l+1)!}e^{-\lambda_2 t}b(t)dt\\ &+ \int_0^\infty (\lambda_1 t)e^{-\lambda_1 t}b^*(t)dt \int_0^\infty \frac{(\lambda_2 t)^{j-l}}{(j-l)!}e^{-\lambda_2 t}b^*(t)dt.\end{aligned}$$

Case 3 : $\mathbf{X_n^1 = 0},\ \mathbf{X_n^2 = 0}$

We introduce two possibility according to the nature of first arrival:

∗If $\mathbf{A}'$:" the first arriving is a priority request": in this case t_{n+1} is instant of the end of service of the priority request, at that time the $(X_{n+1}^1,\ X_{n+1}^2)$ is given by:$\begin{cases} X_{n+1}^1 = A_{n+1}, \\ X_{n+1}^2 = A_{n+1}^2. \end{cases}$

∗If $\mathbf{B}'$:" the first arriving is a non-priority request": in this case t_{n+1} is either, " the end of service of non-priority request or instant of interruption of priority request ".

If t_{n+1} is the end of service of non-priority request, at that time the $(X_{n+1}^1,\ X_{n+1}^2)$ is given by:$\begin{cases} X_{n+1}^1 = 0, \\ X_{n+1}^2 = A_{n+1}^2. \end{cases}$

If t_{n+1} is instant of interruption of priority request, at that time the sequence $(X_{n+1}^1,\ X_{n+1}^2)$ is given by : $\begin{cases} X_{n+1}^1 = 1, \\ X_{n+1}^2 = A_{n+1}^2. \end{cases}$

Therefore the probability of transition $P_{k,l}(i,j)$is:

$$\begin{aligned}P_{k,l}(i,j) = \frac{\lambda_1}{\lambda_1+\lambda_2}\int_0^\infty \frac{(\lambda_1 t)^i}{i!}\frac{(\lambda_2 t)^j}{j!}e^{-(\lambda_1+\lambda_2)t}b(t)dt + \frac{\lambda_2}{\lambda_1+\lambda_2} \times\\ \times \left[\int_0^\infty e^{-\lambda_1 t}b(t)dt \int_0^\infty \frac{(\lambda_2 t)^j}{j!}e^{-\lambda_2 t}b(t)dt + \int_0^\infty (\lambda_1 t)e^{-\lambda_1 t}b^*(t)dt \int_0^\infty \frac{(\lambda_2 t)^j}{j!}e^{-\lambda_2 t}b^*(t)dt.\right]\end{aligned}$$

The probability of realization of A' and B' are respectively :
$P(A') = \frac{\lambda_1}{\lambda_1+\lambda_2}, P(B') = \frac{\lambda_2}{\lambda_1+\lambda_2}$.

We consider at the same time the $M_2/G/1$ queueing system with preemptive priority when $\lambda_1 = 0$.

$\hat{X_n^1}$: the number of priority request in the system just after the end of the n^{th} service or just before interruption.
$\hat{X_n^2}$: the number of non-priority request in the system at the end of the n^{th} service or just before interruption.

The transition operator $\hat{P}_{k,l}(i,j)$ is given by :

$$\hat{P}_{k,l}(i,j) = \begin{cases} \bullet \int_0^\infty \frac{(\lambda_2 t)^{j-l}}{(j-l)!} e^{-\lambda_2 t} b(t)dt, ; if\ i = k-1,\ j \geq l,\ k \neq 0,\ l \geq 0, \\ \bullet \int_0^\infty \frac{(\lambda_2 t)^{j-l+1}}{(j-l+1)!} e^{-\lambda_2 t} b(t)dt, if\ 1 \leq l \leq j+1,\ k = 0, i = 0,\ l \neq 0, \\ \bullet \int_0^\infty \frac{(\lambda_2 t)^{j}}{j!} e^{-\lambda_2 t} b(t)dt, if\ i \geq 0,\ j \geq 0,\ k = 0,\ l = 0, \\ \bullet\ 0 \quad otherwise. \end{cases}$$

To estimate the difference between the stationary distribution of the chain $X_n = (X_{n+1}^1,\ X_{n+1}^2)$ in the $M_2/G/1$ and $\hat{X}_n = (\hat{X}_{n+1}^1,\ \hat{X}_{n+1}^2)$ in the $M/G/1$ system, we apply the strong stability criterion.

3 Strong Stability in an $M_2/G/1$ Queue with Preemptive Priority

In this section, we determine the domain within the system $M_2/G/1$ is strongly v-stable after a small perturbation of the intensity of the priority flux.

Theorem 1. *Let us denote* $\beta_0 = \sup(\beta :\ \hat{f}(\lambda\beta - \lambda) < \beta)$

- $\lambda_2\ E(U) < 1$
- $\exists\ a > 0$ *such as* $E(e^{aU}) = \int_0^\infty\ e^{au}\ b(u)\ du\ < \infty$.

Where introduce the following condition of ergodicity, then for all β *such that* $1 < \beta \leq a$, *the imbedded Markov chain* $\hat{X}_n = (\hat{X}_n^1,\ \hat{X}_n^2)$ *is strongly stable for the function* $v(i,j) = a^i\beta^j$, *where* $\alpha = \frac{\hat{f}(\lambda_2\beta - \lambda_2)}{\rho}$, $\rho = \frac{\hat{f}(\lambda_2\beta - \lambda_2)}{\beta} < 1$ *and* $\hat{f}(\lambda_2\beta - \lambda_2) = \int\limits_0^\infty e^{(\lambda_2\beta - \lambda_2)u} b(u)du$.

Proof. To be able to prove the v-stability of the $M/G/1$ queue with priority we choose:
$V(i,\ j) = a^i \times \beta^j,\ \alpha > 1,\ \beta > 1.\quad h(k,\ l) = 1_{(k=0,\ l=0)}$
and $\alpha(i,\ j) = \hat{P}_{0,0}(0,j)$; where $\hat{P}_{0,0}(0,j) = \int_0^\infty \frac{(\lambda_2 t)^j}{j!} e^{-\lambda_2 t} b(t)dt$

We apply theorem [1]:

$$\hat{\pi}h = \sum_{k\geq 0}\sum_{l\geq 0} \hat{\pi}_{k,l}(i,j)h_{k,l}\ = \hat{\pi}_{0,0}\ = 1 - \frac{\lambda_2}{\mu}\ > 0.$$

$$\alpha 1 = \sum_{i\geq 0}\sum_{j\geq 0} \alpha(i,j) = \sum_{j\geq 0} \int_0^\infty \frac{(\lambda_2 t)^j}{j!} e^{-\lambda_2 t} b(t)dt = \int_0^\infty b(t)dt = 1.$$

$$\alpha h = \sum_{k\geq 0}\sum_{l\geq 0} \alpha_{k,l}h_{k,l} = \alpha_{0,0}(i,\ j) = \hat{P}_{0,0}(0,\ j) > 0.$$

Verification of a. We have two cases: $\mathbf{k = 0},\ \mathbf{l = 0}$

$$T_{0,0}(i,j) = \hat{P}_{0,0}(0,j) - 1\hat{P}_{0,0}(0,j) = 0 \geq 0.$$

k = 0, l ≠ 0, k ≠ 0, l = 0, k ≠ 0, l ≠ 0

$$T_{k,l}(i,\ j) = \hat{P}_{k,l}(i,j) - 0\hat{P}_{0,0}(0,\ j) = \hat{P}_{k,l}(i,\ j) \geq 0$$

Verification of b. We have three cases:
1) k=0, l= 0

$$TV(0,0) = \sum_{i\geq 0}\sum_{j\geq 0}\alpha^i\beta^j T_{0,0}(i,j) = 0\ \leq\ a^0\beta^0\rho = \rho$$

2) k=0, l ≠ 0

$$TV(0,l) = \sum_{j\geq 0}\beta^j \int_0^\infty \frac{(\lambda_2 t)^{j-l+1}}{(j-l+1)!}e^{-\lambda_2 t}b(t)dt \leq \beta^l\ \hat{f}(\lambda_2\beta - \lambda_2)$$

It is sufficient to verify :

$$\beta^{l-1}\hat{f}(\lambda_2\beta - \lambda_2) \leq \rho\beta^l \Longleftrightarrow \rho\beta \geq \hat{f}(\lambda_2\beta - \lambda_2)$$

3) k ≠ 0, l ≠ 0

$$TV(k,l) = \sum_{i\geq 0}\sum_{j\geq 0} T_{k,l}(i,j)v(i,j) = a^{k-1}\beta^j \sum_{j\geq l}\hat{P}_{k-1,l}(i,j)$$

$$= \beta^l a^{k-1}\int_0^\infty \sum_{j\geq l}\frac{(\lambda_2 t\beta)^{j-l}}{(j-l)!}e^{-\lambda_2 t}b(t)dt = \beta^l a^{k-1}\hat{f}(\lambda_2\beta - \lambda_2)$$

It is sufficient to verify:

$$a^{k-1}\ \beta^l\ \hat{f}(\lambda_2\alpha - \lambda_2)\ \leq\ \rho\ a^k\ \beta^l \Longleftrightarrow \rho\ a \geq\ \hat{f}(\lambda_2\beta - \lambda_2)$$

Therefore we must verify : $\begin{cases} \rho\ \beta\ \geq \hat{f}(\lambda_2\beta - \lambda_2), \\ \rho a\ \geq\ \hat{f}(\lambda_2\beta - \lambda_2) \end{cases}$ Let us choose :

$$\rho = \frac{\hat{f}(\lambda_2\beta - \lambda_2)}{\beta} < 1. \tag{1}$$

Then for all $1 < \beta \leq \beta_0,\ a \geq \beta,\ \rho < 1$, We have, $TV(k,l) \leq \rho\ V(k,l)$.

Finally we verify the condition (c):

$$T_{k,l}(i,j) = \hat{P}_{k,l}(i,\ j) - h(k,l)\alpha(k,\ l) \Longrightarrow \hat{P}_{k,l} = T_{k,l}(i,\ j) + h(k,\ l)\alpha(k,\ l)$$
$$\Longrightarrow \|\hat{P}_{k,l}\|_v = ||T_{k,l}(i,\ j) + h(k,\ l)\alpha(k,\ l)||_v \leq \|T_{k,l}(i,\ j)\|_v + ||h(k,\ l)||_v\ \|\alpha(k,\ l)\|_v$$

$$\|T_{k,l}(i,\ j)\|_v = \leq\ \sup_{k\geq 0}\sup_{l\geq 0}\ \frac{1}{V(k,\ l)}\ \rho\ V(k,\ l)\ \leq \rho\ < 1$$

Because ,

$$\sum_{i\geq 0}\sum_{j\geq 0} V(i,\ j)T_{k,l}(i,\ j) = \|TV(k,\ l)\|_v\ \leq \rho\ V(k,\ l).$$

$$||h(k,l)||_v = \sup_{k\geq 0}\sup_{l\geq 0}\frac{|h(k,l)|}{V(k,\ l)} = 1$$

$$||\alpha(k,l)||_v = \sum_{j\geq 0} a^0\ \beta^j\ \hat{P}_{0,0}(0,j) = \sum_{j\geq 0}\ \int_0^\infty\ \frac{(\lambda_2\beta t)^j}{j!}\ e^{-\lambda_2 t}\ b(t)\ dt = \hat{f}(\lambda\beta-\lambda)$$

Therefore,$||\hat{P}_{k,l}|| \leq 1 + \hat{f}(\lambda\beta-\lambda) < \infty$.

3.1 Estimation of Stability

In order to obtain the error due to the approximation of the system $M_2/G/1$ by the $M/G/1$ one, let us estimate the norm of deviation of the transition kernel.

Estimation of Deviation of Transition Kernels
To estimate the margin between the stationary distribution of Markov chain $\hat{X}_n$ and X_n, first we estimate the norm of the deviation of transition kernels.

Theorem 2. *For all β and α, such as $1 < \beta \leq \alpha$, $|| \triangle ||_v = ||\ P - \hat{P}\ ||_v \leq D$. Such that,*

$$D = \max\{K_1, K_2, K_3\} \tag{2}$$

$$K_1 = \frac{\lambda_1}{\lambda_1+\lambda_2}\int_0^\infty e^{(\alpha-1)\lambda_1 t}e^{(\beta-1)\lambda_2 t}b(t)dt + \frac{\lambda_2}{\lambda_1+\lambda_2}\int_0^\infty e^{-\lambda_1 t}b(t)dt\int_0^\infty e^{(\beta-1)\lambda_2 t}b(t)dt$$
$$+\ \frac{\lambda_2}{\lambda_1+\lambda_2}\int_0^\infty \lambda_1 t e^{-\lambda_1 t}b^*(t)dt\int_0^\infty e^{(\beta-1)\lambda_2 t}b^*(t)dt$$

$$K_2 = \frac{1}{\beta}\int_0^\infty e^{-\lambda_1 t}b(t)dt\int_0^\infty e^{(\beta-1)\lambda_2 t}b(t)dt + \int_0^\infty \lambda_1 t e^{-\lambda_1 t}b^*(t)dt\int_0^\infty e^{(\beta-1)\lambda_2 t}b^*(t)dt$$

$$K_3 = \frac{1}{\alpha}\left[\int_0^\infty e^{(\alpha-1)\lambda_1 t+(\beta-1)\lambda_2 t}b(t)dt + \int_0^\infty e^{\lambda_2(\beta-1)t}b(t)dt\right]$$

We must choose smallest of the estimates obtained in $\{K_1, K_2, K_3,\}$.

These intermediate results allow us to consider the problem of obtaining estimates of stability, with an exact computation of the constants. for this, we introduce:

$\pi(i,j)$: the joint stationary distribution of the process of number of request of the priority and non-priority of the system $M_2/G/1$.

$\hat{\pi}(i,j)$: the joint stationary distribution of the process of the number of the non-priority request of the system $M/G/1$

Generating Function

In order to estimate the norm $\|\hat{\pi}\|_v$, necessary for the obtaining of the stability inequalities, let us calculate the generating function $\Pi(Z_1, Z_2)$ of $\hat{\pi}$.

Theorem 3. *Let us note $\Pi(Z_1, Z_2)$ the generating function of the $\hat{\pi}(i,j)$ (stationary distribution of the $M/G/1$ system).If the two conditions of ergodicity are verified:*

$$\begin{cases} \lambda_2 E(U) \leq 1 \\ \exists\ a > 0,\ suchthat\ E(e^{aU}) = \int_0^\infty e^{au} b(u)du < \infty. \end{cases}$$

We have the equality: $\Pi(Z_1, Z_2) = \frac{(Z_2-1)\hat{f}(\lambda_2 Z_2 - \lambda_2)}{Z_2 - \hat{f}(\lambda_2 Z_2 - \lambda_2)}(1 - \frac{\lambda_2}{\mu})$.

Where

$$\hat{f}(\lambda_2 Z_2 - \lambda_2) = \int_0^\infty e^{\lambda_2(Z_2-1)t}\ b(t)dt. \tag{3}$$

and

$$\mu = E(u) = \int_0^\infty t\ b(t)\ dt. \tag{4}$$

Proof. $\Pi(Z_1, Z_2) = \sum_{i\geq 0} Z_1^i \Pi(i, Z_2) = \Pi(0, Z_2) + \sum_{i\geq 1} Z_1^i \Pi(i, Z_2)$

Where $\Pi(i, Z_2) = \sum_{j\geq 0} \hat{\pi}(i,j) Z_2^j$.

From the transition of the transition kernel, we have:

$$\hat{\pi}(i,j) = \sum_k \sum_l \hat{\pi}(k,l)\hat{P}_{k,l}(i,j) = 1_{i=0}\ \hat{\pi}(0,j)\hat{P}_{0,0}(0,j) + 1_{i=0} \sum_{l>0} \hat{\pi}(0,l)\hat{P}_{0,l}(0,j)$$

$$+1_{j\geq 0} \sum_{k>0} \{\sum_{l=0}^{j} \hat{\pi}(k,l)\hat{P}_{k,l}\}_{1_{i=k-1}}$$

For i = 0

$$\hat{\pi}(0,j) = \hat{\pi}(0,0) \int_0^\infty \tfrac{(\lambda_2 t)^j}{j!} e^{-\lambda_2 t} b(t)dt + \sum_{l=1}^{j+1} \hat{\pi}(0,l) \int_0^\infty \tfrac{(\lambda_2 t)^{j-l+1}}{(j-l+1)!} e^{-\lambda_2 t} b(t)dt$$

$$+ \sum_{l=0}^{j} \hat{\pi}(1,l) \int_0^\infty \tfrac{(\lambda_2 t)^{j-l}}{(j-l)!} e^{-\lambda_2 t} b(t)dt.$$

Using the method of generating functions, we obtain:

$$\Pi(1, Z_2) = \Pi(0, Z_2) \frac{Z_2 - \hat{f}(\lambda_2 Z_2 - \lambda_2)}{Z_2 \hat{f}(\lambda_2 Z_2 - \lambda_2)} + \frac{1 - Z_2 \hat{f}(\lambda_2 Z_2 - \lambda_2)}{Z_2 \hat{f}(\lambda_2 Z_2 - \lambda_2)} \hat{\pi}(0,0)$$

For i > 0

$$\hat{\pi}(i,j) = \sum_{l\geq 0} \hat{\pi}(k,l) \int_0^\infty \tfrac{(\lambda_2 t)^{j-l}}{(j-l)!} e^{-\lambda_2 t} b(t)dt.$$

And,

$$\Pi(i, Z_2) = \Pi(i+1, Z_2)\hat{f}(\lambda_2 Z_2 - \lambda_2) \Longleftrightarrow \Pi(i, Z_2) = \frac{\Pi(1, Z_2)}{\hat{f}^{i-1}(\lambda_2 Z_2 - \lambda_2)}$$

If there is no priority request in the system (when $\theta = 0$), we are in case of the system $M/G/1$.

$$\Pi(0, Z_2) = \frac{(Z_2 - 1)\hat{f}(\lambda_2 Z_2 - 1)}{Z_2 - \hat{f}(\lambda_2 Z_2 - \lambda_2)}\hat{\pi}(0, 0) \tag{5}$$

This is the Pollatchek- Khinchin formula.

3.2 Inegality of Stability

In the following theorem, we calculate the error due to approximate $M_2/G/1$ system by the $M/G/1$ one on the stationary distribution.

Theorem 4. *Suppose that in a system $M_2/G/1$ with preemptive priority, the conditions of theorem (1) hold. Then, $\forall\beta$ and α, $1 < \beta \leq \alpha$,; we have the estimation*

$$\|\pi - \hat{\pi}\| \leq W_\theta, \tag{6}$$

Where,
$W_\theta = D(1 + W)W(1 - \rho - (1 + W)D)^{-1}$,
$W = (\beta - 1)(1 - \lambda_2/\mu)\frac{\rho}{1-\rho}$,
$D = \min\{K_1, K_2, K_3\}$.
and ρ, μ are respectively defined in (1),(4)

Proof. To verify the theorem (4), it is sufficient to estimate $\|\pi\|_v$ and $\|1\|_v$, where 1 is the function identically equal to unity. $\|\hat{\pi}\|_v = \sum\limits_{i\geq 0}\sum\limits_{j\geq 0} v(i,j)|\hat{\pi}(i,j,)|$, Where $v(i,j) = a^i\beta^j$.

From (5), we have, $||\hat{\pi}||_\nu = \frac{(\beta-1)\hat{f}(\lambda_2\beta-\lambda_2)}{\beta-\hat{f}(\lambda_2\beta-\lambda_2)}(1 - \frac{\lambda_2}{\mu}) = W$

$||1||_\nu = \sup\limits_{k\geq 0}\sup\limits_{l\geq 0}\frac{1}{a^k\beta^l} \leq 1$.

By definition, $C = 1 + ||\ I\ ||_v||\ \hat{\pi}\ ||_v = 1 + W$. And, $||\ \triangle\ ||_\nu < \frac{1-\rho}{C}$.

Thence,

$||\ \pi - \hat{\pi}\ ||_v = D(1 + W)W(1 - \rho - (1 + W)D)^{-1}$.

4 Conclusion

In this work, we are obtained the measurement and performance of the systems of queues with preemptive priority. We were interested in the study of strong stability in a system $M_2/G/1$ with preemptive priority, after perturbation of the intensity of the arrivals of the priority requests. We clarified the conditions of approximation of the characteristics of the system of queue $M_2/G/1$ with preemptive priority by those corresponding to the system of queue $M/G/1$ classical. The method of strong stability also makes it possible to obtain the quantitative estimates of stability. We obtained the inequalities of stability with an exact calculation of the constants.

References

1. Aissani, D., Kartashov, N.V.: Ergodicity and stability of Markov chains with respects of operator topology in the space of transition kernels. Dokl. Akad. Nauk. Ukr. S.S.R, Ser. A 11, 3–5 (1983)
2. Anisimo, V.V.: Estimates for the deviations of the transition characteristics of non-homogeneous Markov processes. Ukrainian Math. Journal 40, 588–592 (1986)
3. Borovkov, A.A.: Asymptotic methods in queueing theory. Wiley, New York (1984)
4. Franken., P.: Ein stetigkeitssatz fur verlustsysteme. Operations-Forschung und Math., Stat. 11, 1–23 (1995)
5. Gnedenko, B.V., Klimov, G.P., Danelian, E.: Priority queueing systems. Moskow University, Moskow (1973)
6. Ipsen, C.F., Meyer, D.: Uniform stablity of Markov chains. SIAM Journal on Matrix Analysis and Applications 15(4), 1061–1074 (1994)
7. Jaiswal, N.K.: Priority Queues. Academic Press, New York (1968)
8. Kalashinkov, V.V.: Quantitative estimates in queueing theory. CRC Press, Inc., Boca Raton (1996)
9. Kartashov, N.V.: Strong stable Markovs chains. VSP Utrecht (1996)
10. Kennedy, D.P.: The continuity of single server queue. Journal of Applied Probability 09, 370–381 (1972)
11. Rachev, S.: T The problem of stability in queueing theory. Queueing Systems 4, 287–318 (1989)
12. Rosseberg, H.J.: Uber die verteilung von watereiten. Mathematische Nachrichten 30, 1–16 (1965)
13. Stoyan, D.: Ein stetigkeitssatz fur einlinge wartemodelle eer bedienungsheorie. Maths Operations forschu. Statist. (3), 103–111 (1977)
14. Zolotariev, V.M.: On the continuity of stochastic sequences generated by recurrent processes. Theory of Probabilities and its Applications 20, 819–832 (1975)

Outcome-Space Polyblock Approximation Algorithm for Optimizing over Efficient Sets*

Bach Kim Nguyen Thi[1], Hoai An Le Thi[2], and Minh Thanh Tran[1]

[1] Faculty of Applied Mathematics and Informatics
Hanoi University of Technology, Vietnam
kimntb-fami@mail.hut.edu.vn
[2] Laboratory of Theoretical and Applied Computer Science - LITA EA 3097
UFR MIM, University of Paul Verlaine - Metz, Ile de Saulcy, 57045 Metz, France
lethi@sciences.univ-metz.fr

Abstract. We propose an outcome-space polyblock approximation algorithm for maximizing a function $f(x) = \varphi(Cx)$ over the efficient solution set X_E of the multiple objective linear programming problem Max $\{Cx|x \in X\}$. The convergence of the algorithm is established. To illustrate the new algorithm, we apply it to the solution of a sample problem.

Keywords: Increasing function; Polyblock approximation algorithm; Multiple objective linear programming; Optimization over the efficient set.

1 Introduction

This paper is concerned with the problem of optimizing

$$\max \ f(x), \quad \text{s.t.} \ \ x \in X_E, \tag{P}$$

where f is a real valued function and X_E is the efficient solution set of the multiple objective linear programming problem

$$\text{Max } Cx, \quad \text{s.t.} \ \ x \in X, \tag{VP}$$

where C is a $p \times n$ matrix and $X \subset \mathbb{R}^n$ is a nonempty compact polyhedron. Recall that a point $x^0 \in X$ is called an efficient solution to problem (VP) if there is no point $x \in X$ such that $Cx \geq Cx^0$ and $Cx \neq Cx^0$. The efficient solution set X_E consists of some closed faces of the polyhedron X. While this set is always pathwise connected, generally, it is not convex [7]. Therefore optimizing over the efficient set is a hard task.

Problem (P) has many applications in decision making and have attracted a great deal of attention from researchers (see e.g. [1], [2],[3], [4],[6], [8], [9], [10]

* Supported in part by the National Basic Program of Natural Science, Vietnam and another part by "Agence Universitaire pour la Francophonnie" (AUF).

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 234–243, 2008.

and references therein). In many practical problems such functions f have been constructed in the form depending on the criteria of Problem (VP), $f(x) = \varphi(Cx)$ with a function $\varphi : \mathbb{R}^p \to \mathbb{R}$. Isermann and Steuer published [6] the cutting plane algorithm solving (P) with $f(x) = -\langle c^i, x\rangle$ where c^i is the i^{th} row of C. Benson [3] has studied (P) for the case when φ is a linear function, i.e. the function $f(x)$ is linearly dependent upon the rows of C. Muu and Luc [8] proposed inner-approximation algorithms for solving (P) with convex functions φ. Kim and Muu in [10] studied the efficient set X_E for Problem (VP) by using its projection into the linear space spanned by the independent criteria and proposed a simple algorithm for solving (P) with φ being a convex function.

In this paper we focused on a special class of (P) noted (P_1)

$$\max \varphi(Cx), \quad \text{s.t.} \quad x \in X_E, \tag{P_1}$$

when φ is a continuous and increasing function defined on $\mathbb{R}^p_+$. Note that a function $\varphi : \mathbb{R}^p_+ \to \mathbb{R}$ is *increasing* on $\mathbb{R}^p_+$ if for $y', y \in \mathbb{R}^p_+$ and $y' \geq y$ we have $\varphi(y') \geq \varphi(y)$.

Assume throughout this paper that the nonempty compact polyhedron X is defined by

$$X = \{x \in \mathbb{R}^n | Ax = h, x \geq 0\}, \tag{1}$$

where A is a $m \times n$ matrix and $h \in \mathbb{R}^m$, and the *outcome set* Y,

$$Y := \{y \in \mathbb{R}^p \mid y = Cx \ \text{ for some } \ x \in X\},$$

is a subset of the interior of $\mathbb{R}^p_+ := \{y \in \mathbb{R}^p \mid y \geq 0\}$,

$$Y \ \subset \text{int}\mathbb{R}^p_+. \tag{2}$$

From [11] and [7], Y is a nonempty, compact polyhedron and X_E is nonempty. The requirement (2) can be achieved by considering the criteria functions $(Cx - y^M)$ instead of Cx, where $y^M = (y_1^M, \ldots, y_p^M)$ and

$$y_i^M < y_i^* = \min\{y_i, y \in Y\}, \ i = 1, \ldots, p.$$

The outcome-space reformulation of problem (P_1) is given by the maximizing a continuous and increasing function in the outcome-space $\mathbb{R}^p$,

$$\max \varphi(y), \ \text{s.t.} \ y \in Y_E, \tag{P_2}$$

where $Y_E := \{y \in \mathbb{R}^p \mid y = Cx \ \text{ for some } \ x \in X_E\}$. It is easily seen that if $\bar{y}$ is a solution to problem (P_2) then any $\bar{x} \in X$ such that $C\bar{x} = \bar{y}$ is an optimal solution to problem (P_1).

Here, we present an outcome-space polyblock approximation algorithm for solving problem (P_1). Instead of solving problem (P_1), we solve problem (P_2). Fortunately, the dimension p of the outcome space is typically much smaller than the dimension n of the decision space. Therefore, computational savings could be obtained.

The bases of the algorithm are presented in the next section. The algorithm and its convergence are discussed in Section 3.

2 Bases of the Algorithm

2.1 Equivalent Form of Problem ($\mathbf{P_2}$)

For a given nonempty set $Q \subset \mathbb{R}^p$, a point $q^0 \in Q$ is an *efficient point* (or *Pareto point*) of Q if there is no $q \in Q$ satisfying $q \geq q^0$ and $q \neq q^0$, i.e. $Q \cap (q^0 + \mathbb{R}^p_+) = \{q^0\}$. Similarly, a point $q^0 \in Q$ is a *weakly efficient point* if there is no $q \in Q$ satisfying $q \gg q^0$, i.e. $Q \cap (q^0 + \text{int}\mathbb{R}^p_+) = \emptyset$. We denote by Max$Q$ and WMax$Q$ the set of all efficient point of Q and the set of all weakly efficient point of Q, respectively. By the definition,

$$\text{Max}Q \subseteq \text{WMax}Q.$$

For $b \in \mathbb{R}^p_+$, we denote by $[0, b] := \{y \in \mathbb{R}^p \mid 0 \leq y \leq b\}$ the *box* (or the hyper-rectangle) response to vertex b.

Consider the set $N(Y)$ defined by

$$N(Y) := \bigcup_{y \in Y} [0, y] = (Y - \mathbb{R}^p_+) \cap \mathbb{R}^p_+,$$

where $Y = \{y \in \mathbb{R}^p \mid y = Cx \quad \text{for some} \quad x \in X\}$. It is clear that $N(Y)$ is a nonempty, full-dimension bounded polyhedron in $\mathbb{R}^p_+$.

Proposition 1. i) Max$N(Y)$ = Max$Y$;

$$\text{ii)} \quad N(Y) = \bigcup_{y \in \text{Max}Y} [0, y].$$

Proof. i) Since $N(Y) = (Y - \mathbb{R}^p_+) \cap \mathbb{R}^p_+$, we have

$$Y \subseteq N(Y) \subseteq Y - \mathbb{R}^p_+. \tag{3}$$

First, we prove that Max$Y \subseteq$ Max$N(Y)$. Let $y^e \in$ Max$Y$. From (3) it follows that $y^e \in N(Y)$. Assume on the contrary that $y^e \notin$ Max$N(Y)$. Hence, there is $\bar{y} \in N(Y)$ such that $\bar{y} > y^e$. Furthermore, since $\bar{y} \in N(Y) \subseteq Y - \mathbb{R}^p_+$, we have $\bar{y} = y^0 - v$, where $y^0 \in Y$ and $v \geq 0$. Hence $y^0 > y^e$ which conflicts with the hypothesis $y^e \in$ Max$Y$. So $y^e \in$ Max$N(Y)$.

Now, we show that Max$N(Y) \subseteq$ Max$Y$. Let $y^e \in$ Max$N(Y)$. To prove $y^e \in$ Max$Y$, we need to show only that $y^e \in Y$, because of $Y \subset N(Y)$. Similarly to above arguments, we have $y^e = y^0 - v$ with $y^0 \in Y$ and $v \geq 0$. This implies that $v = 0$. Indeed, if $v > 0$, we have that $y^0 > y^e$ that contradicts to the fact $y^e \in$ Max$N(Y)$. Thus, $y^e \in Y$.

ii) By the definition of $N(Y)$ we have

$$\bigcup_{y \in \text{Max}Y} [0, y] \subseteq N(Y).$$

We need only to show that

$$\bigcup_{y \in \text{Max}Y} [0, y] \supseteq N(Y).$$

As Y is compact, Max$Y$ is nonempty ([7]). From the definition of Max$Y$ we can see easily that if $y^e \in \text{Max}Y$, then there is no point $y \in Y$ such that $[0, y^e] \subset [0, y]$. The proof is straightforward.

Proposition 2. *Problem (P_2) is equivalent to the problem*

$$\max \varphi(y), \quad \text{s.t.} \quad y \in \text{WMax}N(Y). \tag{P_3}$$

Proof. First, we will show that Problem (P_2) is equivalent to the following problem

$$\max \varphi(y), \quad \text{s.t.} \quad y \in \text{Max}N(Y). \tag{P_3^*}$$

From (i) of Proposition 1 we have Max$N(Y) =$ Max$Y$. We need only to show that Max$Y = Y_E$. Indeed, a point $y^0 \in \text{Max}Y$ if $y^0 \in Y$ and there is no $y \in Y$ such that $y > y^0$. It means that there is a point $x^0 \in X$ such that $y^0 = Cx^0$ and there is no $x \in X$ such that $y = Cx > y^0 = Cx^0$. In other words, $x^0 \in X_E$, i.e. $y^0 \in Y_E$.

Now, by the above observation we will show that (P_3) is equivalent to (P_3^*). As Max$N(Y) \subseteq$ WMax$N(Y)$, we need only to prove that if y^0 is an optimal solution of (P_3), then $y^0 \in \text{Max}N(Y)$. Assume on the contrary that $y^0 \in \text{WMax}N(Y) \setminus \text{Max}N(Y)$. Consider the cone $y^0 + \mathbb{R}^p_+$. By definition and the pathwise connectedness of the weakly efficient point set WMax$N(Y)$, there is $y^e \in \text{Max}N(Y)$ such that

$$y^e = y^0 + te^{i^0} \text{ for some } i^0 \in \{1, \ldots, p\} \text{ and } t \geq 0.$$

As $y^e \geq y^0$, we have $\varphi(y^e) \geq \varphi(y^0)$, since the function φ is increasing. This contradicts the fact that y^0 is an optimal solution of (P_3).

Next, we present some estimations which serve as a basis of the procedure in our algorithm to solve the problem (P_3).

For $i = 1, \ldots, p$ we denote $F_i := \{y \in N(Y) | y_i = 0\}$. Obviously, the sets F_i are $(p-1)$-dimensional faces of $N(Y)$. Recall that WMax$N(Y)$ is a union of some closed faces of $N(Y)$. Since $0 \in F_i$ and $0 \notin \text{WMax}N(Y)$, we have that

$$\text{ri}F_i \cap \text{WMax}N(Y) = \emptyset, i = 1, \ldots, p.$$

The boundary $\partial N(Y)$ of $N(Y)$ can be described as follows.

Proposition 3

$$\partial N(Y) = (\bigcup_{i=1}^{p} F_i) \cup \text{WMax}N(Y).$$

Proof. Observe first that

$$(\bigcup_{i=1}^{p} F_i) \cup \text{WMax}N(Y) \subseteq \partial N(Y).$$

From this we need to show only that if a point $y^* \in \partial N(Y) \setminus (\cup_{i=1}^p F_i)$, then $y^* \in \mathrm{WMax}N(Y)$. Given a point y^* belonging to $\partial N(Y) \setminus (\cup_{i=1}^p F_i)$. Then, y^* is contained in a face F of $N(Y)$. Note that a subset $F \subset N(Y)$ is a face if there is a vector $v \in \mathbb{R}^p$ such that F is the optimal solution set of the linear programming problem

$$\max\langle v, y\rangle, \text{ s.t. } y \in N(Y).$$

Since $Y \subset \mathrm{int}\mathbb{R}^p_+$, $\quad N(Y) = (Y - \mathbb{R}^p_+) \cap \mathbb{R}^p_+$ and $F \neq F_i$ for all $i = 1, \ldots, p$, we have $v \in \mathbb{R}^p$, $v \geq 0$ and $v \neq 0$. It is well known (see for instance Theorem 2.5, Chapter 4 of [7]) that $y^0 \in N(Y)$ is a weakly efficient point if and only if there is a nonzero vector $\lambda \geq 0$ such that y^0 is a maximum point of the function $\langle \lambda, y\rangle$ over $N(Y)$. Therefore, $y^* \in \mathrm{WMax}Y$.

The following corollary is immediate from Proposition 3.

Corollary 1. *Suppose $y^* > 0$ and $y^* \notin N(Y)$. The segment $\{\mu y^* | 0 \leq \mu \leq 1\}$ contains a unique point on the* $\mathrm{WMax}N(Y)$.

Let $b := (b_1, \ldots, b_p)$ where $b_i := \max\{y_i : y \in Y\}, i = 1, \ldots, p$. We have $N(Y) \subseteq [0, b]$. The next fact will play an important role in our algorithm.

Proposition 4. *For any $v \in [0, b] \setminus N(Y)$ the segment $\{\mu v | 0 \leq \mu \leq 1\}$ contains a unique point on the* $\mathrm{Max}N(Y)$.

Proof. By Corollary 1, let y^v be the unique weakly efficient point of $N(Y)$ on the segment $\{\mu v | 0 \leq \mu \leq 1\}$. Assume the contrary that

$$y^v \in \mathrm{WMax}N(Y) \setminus \mathrm{Max}N(Y).$$

This implies that there is $i_0 \in \{1, \cdots, p\}$ such that $y^v_{i_0} = b_{i_0} = \max\{y_{i_0} \mid y \in Y\}$. Since $v \in [0, b] \setminus N(Y)$, we always have $v = \lambda y^v$ with $\lambda > 1$. Hence,

$$v_{i_0} = \lambda y^v_{i_0} = \lambda b_{i_0} > b_{i_0}.$$

This is impossible, because $b_{i_0} \geq v_{i_0}$ for all $v \in [0, b]$.

2.2 Polyblock Approximation

A set of the form $Q = \bigcup_{v \in V}[0, y]$ with $|V| < +\infty$ is called a *polyblock* with *vertex set* V. A vertex $v \in V$ is said to be *proper* if there is no $v' \in V \setminus \{v\}$ such that $[0, v] \subset [0, v']$.

Proposition 5. *The maximum of an increasing function $\varphi(y)$ over a polyblock Q is attained at a proper vertex of Q.*

Proof. For any $y \in Q$, there is a proper vertex v of Q such that $y \in [0, v]$, i.e. $y \leq v$, and therefore $\varphi(y) \leq \varphi(v)$. Since the element number of V is finite, we have

$$v^0 = \mathrm{argmax}\{\varphi(v), v \in V\} = \mathrm{argmax}\{\varphi(y), y \in Q\}.$$

This proves the proposition.

Let $Q^1 = [0, b] \subset \mathbb{R}^p$ where $b := (b_1, \ldots, b_p)$ with

$$b_i := \max\{y_i : y \in Y\}, i = 1, \ldots, p.$$

We have $N(Y) \subseteq [0, b]$. Starting with the box Q^1, the polyblock approximation algorithm will iteratively generate a sequence of polyblocks Q^k, $k = 1, 2, \ldots$, such that

$$Q^1 \supset Q^2 \supset \cdots \supset Q^k \supset \cdots N(Y).$$

In a typical iteration k, the algorithm can be described as follows.

Find

$$v^k \in \operatorname{argmax}\{\varphi(v), v \in V^k\},$$

where V^k is the proper vertex set of the polyblock Q^k.

a) If the maximizer v^k of $\varphi(y)$ over the polyblock Q^k belongs to $N(Y)$, then we obtain a point $\bar{y} := v^k$ and $\bar{y} \in \operatorname{Max} N(Y)$ is an optimal solution of Problem (P_2).

b) Otherwise, we construct a polyblock Q^{k+1} such that

$$Q^k \supset Q^{k+1} \supset N(Y) \text{ and } v^k \notin Q^{k+1}.$$

Determine the proper vertex set V^{k+1} of the polyblock Q^{k+1}. The procedure can be repeated with Q^{k+1}, V^{k+1} in place of Q^k and V^k.

The polyblock Q^{k+1} is constructed by

$$Q^{k+1} = Q^k \setminus [y^k, v^k],$$

where $y^k \in \operatorname{Max} N(Y)$ (see Proposition 4) is the unique common point of the boundary of $N(Y)$ and the open line segment connecting the origin 0 and v^k. To compute the proper vertex set V^{k+1} one can use the following proposition.

Proposition 6. (see [5] or [12]) *Let $D \subset \mathbb{R}^p_+$ be a compact normal set contained in a polyblock M with vertex set V. Let $z \in V \setminus D$ and y be the unique common point of the boundary of D and the open line segment connecting the origin 0 and z. Then $V' = (V \setminus \{z\}) \cup \{x^1, \ldots, x^p\}$ where*

$$x^i = z - (z_i - y_i)e^i, i = 1, \ldots, p \tag{4}$$

and the polyblock M' generated by V' satisfies

$$D \subset M' \subset M, \quad z \in M \setminus M'. \tag{5}$$

3 Polyblock Approximation Algorithm

3.1 Algorithm for Solving Problem (P_2)

Let $e := (1, \ldots, 1) \in \mathbb{R}^p$ and $\epsilon > 0$ be a given sufficient small number. Put

$$N(Y)_\epsilon := \{x \in N(Y) | x \geq \epsilon e\}.$$

In practice we find an approximate optimal solution $y^* \in N(Y)_\epsilon$ such that $\varphi(y^*)$ differs from the maximal value of φ over $N(Y)_\epsilon$ by at most $\eta > 0$. A such solution will be called an (ϵ, η)-*approximate optimal solution* of (P_2) and a solution $y^* \in \operatorname{argmax}\{\varphi(y) | y \in N(Y)_\epsilon\}$ will be called an $\epsilon-$ solution to (P_2). Bellows, we will present an algorithm for finding (ϵ, η)- approximate optimal solution to (P_2).

ALGORITHM

Initialization. Compute $b = (b_1, \ldots, b_p)$, with

$$b_i = \max\{y_i, \quad \text{s.t.} \quad y \in Y\}, \quad i := 1, \ldots, p.$$

If $b \in Y$ **Then** STOP; ($\bar{y} = b$ *is the optimal solution to* (P_2)*. In this case,* y *is an ideal efficient point of* Y.)

Else Set

$Q^1 := [0, b]$, $V^1 = \{b\}$, $\varphi(\bar{y}^0) = -\infty$, $k := 1$;

Go to Iteration k.

Iteration $k, k \geq 1$**.** It consists of five steps.

Step k_1. Find $v^k \in \operatorname{argmax}\{\varphi(v) \ | v \in V^k, v \geq \epsilon e\}$.

Step k_2. **If** $v^k \in N(Y)$ **Then** STOP ($\bar{y} = v^k$ *is an* $\epsilon-$ *optimal solution of* (P_2))

Else find the unique point $y^k \in \partial(N(Y)) \cap (0, v^k)$.

Go to Step k_3.

Step k_3. Let $\bar{y}^k = \operatorname{argmax}\{\varphi(\bar{y}^{k-1}), \varphi(y^k)\}$

If $\varphi(\bar{y}^k) \geq \varphi(v^k) - \eta$ **Then** STOP

($\bar{y} = \bar{y}^k$ *is an* (ϵ, η)*-approximate optimal solution of* (P_2))

Else Goto Step k_4

Step k_4. Set $Q^{k+1} = Q^k \setminus [y^k, v^k]$.

Compute p extreme points of the box $[y^k, v^k]$ that are adjacent to v^k

$$v^{k+1,i} = v^k - (v_i^k - y_i^k)e^i, \quad i = 1, \ldots, p,$$

(e^i is the i^{th} unit vector of $\mathbb{R}^p$). Set

$$V^{k+1} = (V^k \setminus \{v^k\}) \cup \{v^{k+1,1}, \ldots, v^{k+1,p}\}.$$

Step k_5. Set $k = k + 1$ and go to iteration k.

Remark 1. When the algorithm is terminated at an iteration K, we obtain a solution $\bar{y}$ to Problem (P_2). The obtained solution $\bar{y}$ is either an $\epsilon-$ optimal solution to (P_2) or an (ϵ, η)- approximate optimal solution to Problem (P_2).

i) If the solution $\bar{y}$ is an $\epsilon-$ optimal solution to (P_2), this solution must be a vertex v^K of the polyblock Q^K: $\bar{y} = v^K$. Then, to find an optimal solution $\bar{x}$ to Problem (P_1) we need to solve the following system

$$\begin{cases} Cx = \bar{y} \\ Ax = h \\ x \geq 0. \end{cases} \tag{6}$$

ii) In general case, the obtained $\bar{y}$ is only an (ϵ, η)- approximate optimal solution to Problem (P_2). By Proposition 4, we have $\bar{y} \in \text{Max}N(Y)$. Hence, $\bar{y}$ must belong to $\text{Max}Y = Y_E$ and we obtain an optimal solution $\bar{x}$ to (P_1) by solving (6).

Remark 2. Recall that a point $y^I \in Y$ is called an *ideal efficient point* of Y when, $y_i^I = \max y_i$, s.t. $y \in Y$ for $i = 1, 2, \ldots, p$. If Y has an ideal efficient point y^I, then $\text{Max}N(Y) = \text{Max}Y = \{y^I\}$. Thus, if in the initial step of the algorithm we get a solution $y = b \in Y$, this one is just an optimal solution to (P_2).

Remark 3. To find a point $y^k \in \partial(N(Y)) \cap (0, v^k)$ we solve the linear programming problem

$$t_k = \max t$$

$$\begin{cases} 0 < t < 1 \\ Cx - v = tv^k \\ Ax = h \\ x \geq 0, v \geq 0. \end{cases}$$

The obtained solution has the form $y^k = t_k v^k$ that belongs to $\partial(N(Y))$.

3.2 Convergence of Algorithm

Convergence Theorem. *The algorithm terminates after finitely many steps, yielding an $\epsilon-$ optimal solution or an (ϵ, η)- approximate optimal solution to (P_2).*

Proof. For convenience, we will equip $\mathbb{R}^p$ with the normal $\|x\| := \max_i |x_i|$. We will prove that for any $\delta > 0$ there exists a number $k > 0$ such that $\|v^k - y^k\| < \delta$. Note that the function φ is uniformly continuous on the compact set Q^1. Hence, if δ is chosen so that $\varphi(v^k) - \varphi(y^k) < \eta$ whenever $\|v^k - y^k\| < \delta$ for a sufficient large k we will have

$$\varphi(v^k) - \varphi(\bar{y}^k) < \varphi(v^k) - \varphi(y^k) < \eta$$

and the algorithm terminates.

Observe that there are positive numbers $M > m > 0$ independent of k such that

$$M\|v^k - y^k\| \geq (v_i^k - y_i^k) \geq m\|v^k - y^k\|, \ i = 1, \ldots, p$$

since $v^k, y^k \in \{x \in Q^1 : x \geq \epsilon e\}$ and $y^k = \lambda v^k$ for $\lambda \in (0, 1)$. This implies that

$$M^p\|v^k - y^k\|^p \geq Vol([y^k, v^k]) \geq m^p\|v^k - y^k\|^p. \tag{7}$$

Here, $Vol([y^k, v^k])$ indicates the volume of the box $[y^k, v^k]$.

On the other hand, since the boxes $[y^k, v^k]$ are disjoint and all of them are contained in the box Q^1, one can see that

$$\sum_{k=1}^{K} Vol([y^k, v^k]) \leq Vol(Q^1).$$

Hence, for a number k large enough we have $Vol([y^k, v^k]) < (m\delta)^p$. Then, by (7) we get $\|v^k - y^k\| < \delta$.

3.3 Example

To illustrate the application of the outcome-space polyblock approximation algorithm for solving problem (P_1), consider the case with $p = 2$,

$$C = \begin{pmatrix} 2 & -4 & -1 & 0 & -6 & 6 & 7 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 4 & 3 & -4 & 1 & 0 & 2 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 0 & 4 & 0 & 1 & -3 & 6 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 7 & 1 & 8 & 8 & 7 & 0 & 3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 2 & 5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 5 & 7 & -3 & 5 & 2 & 0 & 8 & 5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 & 4 & 0 & 8 & -3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 8 & 0 & 0 & 1 & -3 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$h = (6 \;\; 10 \;\; 5 \;\; 6 \;\; 8 \;\; 10 \;\; 10 \;\; 10)^T.$$

• Solve Problem (P_2) with $\varphi(y) = y_1$, $\epsilon = 0.01$ and $\eta = 0.01$. After 7 iterations, we receive (ϵ, η) - approximate optimal solution $y = (14.493, 9.357) \in Y_E$.

• Solve Problem (P_2) with $\varphi(y) = y_1^{0.25} y_2^{0.75}$, $\epsilon = 0.01$ and $\eta = 0.01$. After 108 iterations, we receive (ϵ, η)-optimal solution $y = (10.616, 5.606) \in Y_E$.

• Solve Problem (P_2) with $\varphi(y) = y_1 + y_2$, $\epsilon = 0.01$ and $\eta = 0.01$. After 8 iterations, we receive (ϵ, η)-optimal solution $y = (14.493, 9.357) \in Y_E$.

References

1. An, L.T.H., Tao, P.D., Muu, L.D.: D.C. optimization approach for optimizing over the efficient set. Operations Research Letters 19, 117–128 (1996)
2. An, L.T.H., Tao, P.D., Thoai, N.V.: Combination between global and local methods for solving an optimization problem over the efficient set. European Journal of Operational Research 142, 258–270 (2002)
3. Benson, H.P.: A Bisection-extreme point search algorithm for optimizing over the efficient set in the linear dependence case. J. of Global Optimization 3, 95–111 (1993)

4. Ecker, G.L., Song, J.H.: Optimizing a Linear Function over a Efficient Set. Journal of Optimization Theory and Applications 83, 541–563 (1994)
5. Tuy, H.: Normal Sets, Polyblocks, and Monotonic Optimization. Vietnam Journal of Mathematics 27, 277–300 (1999)
6. Isermann, H., Steuer, R.E.: Computational experience concerning payoff table and minimum criterion values over the efficient set. European J. of Operations Research 33, 91–97 (1987)
7. Luc, D.T.: Theory of Vector Optimization. Springer, Berlin (1989)
8. Luc, L.T., Muu, L.D.: Global optimization approach to optimizing over the efficient set. In: Gritzmann, P., Host, R., Sachs, E., Tichatschke, R. (eds.) Recent Advances in Optimization, vol. 452, pp. 183–195. Springer, Berlin (1997)
9. Kim, N.T.B.: An algorithm for optimizing over the efficient set. Vietnam Journal of Mathematics 28, 329–340 (2000)
10. Kim, N.T.B., Muu, L.D.: On the projection of the efficient set and potential applications. Optimization 51, 401–421 (2002)
11. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
12. Rubinov, A., Tuy, H.: Heather Mays. An algorithm for monotonic global optimization problems. Optimization 49, 205–221 (2001)

A DC Programming Approach for Mixed-Integer Linear Programs

Yi-Shuai Niu and Tao Pham Dinh

Laboratoire de Mathématiques de l'INSA,
National Institute for Applied Sciences - Rouen,
BP 08, Place Emile Blondel F 76131, Mont Saint Aignan Cedex, France
niuys@insa-rouen.fr, pham@insa-rouen.fr

Abstract. In this paper, we propose a new efficient algorithm for globally solving a class of Mixed Integer Program (MIP). If the objective function is linear with both continuous variables and integer variables, then the problem is called a Mixed Integer Linear Program (MILP). Researches on MILP are important in both theoretical and practical aspects. Our approach for solving a general MILP is based on DC Programming and DC Algorithms. Using a suitable penalty parameter, we can reformulate MILP as a DC programming problem. By virtue of the state of the art in DC Programming research, a very efficient local nonconvex optimization method called DC Algorithm (DCA) was used. Furthermore, a robust global optimization algorithm (GOA-DCA): A hybrid method which combines DCA with a suitable Branch-and-Bound (B&B) method for globally solving general MILP problem is investigated. Moreover, this new solution method for MILP is also applicable to the Integer Linear Program (ILP). An illustrative example and some computational results, which show the robustness, the efficiency and the globality of our algorithm, are reported.

Keywords: MIP, MILP, ILP, DC Programming, DCA, Branch-and-Bound, GOA-DCA.

1 Introduction

The MIP problems are classical discrete optimization problems with both integer and continuous variables. Considering a general formulation of MILP:

$$\begin{array}{ll} \min & f(x,y) := c^T x + d^T y \\ s.t. & Ax + By \leq b, A_{eq}x + B_{eq}y = b_{eq}, \\ & (lb_x, lb_y) \leq (x,y) \leq (ub_x, ub_y), \\ & x \in \mathbb{R}^n, y \in \mathbb{Z}_+^m. \end{array} \tag{1}$$

where the objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a linear function. The variables x (resp. y) are bounded by lb_x and ub_x (resp. lb_y and ub_y) which are constants specifying the lower and upper bounds of x (resp. of y). The difficulty of (1) lies in the variable y, because it is an integer variable which destroys the convexity

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 244–253, 2008.

of the constraint set. If we suppose y is a continuous variable, then (1) becomes to a linear programming problem which can be solved efficiently.

There are several well-known methods for solving MIPs: Branch-and-Bound Method, Cutting-Plane Method, Decomposition Method [7], etc. Some of them have already been implemented in commercial software, such as "ILOG CPLEX", "MATLAB", "XPressMP", "AIMMS", "LINDO/LINGO", "1stOpt", etc. Moreover, there are also open-source codes [11], such as "OSI","CBC" for MILP, and "BONMIN", the latter is a good solver for general Mixed Integer Nonlinear Program (MINLP). Although, we already have several methods and softwares for solving MIPs, it should be emphasized that the research on MIP should be continued, because MIP is classified as NP-hard, for which there is no efficient polynomial time algorithm. The methods mentioned above are often very expensive in computation time. Therefore, there is a need to improve these methods or to find more efficient methods.

In order to overcome the difficulty of the integer variables, we will reformulate the problem (1) to a DC programming problem, then we apply an efficient method for solving DC programming, called DC Algorithm (DCA), which enable us to find a KKT point of the DC reformulation problem. DCA can rapidly generate a convergent sequence on which the objective values decrease.

In order to check the globality of the computed solution obtained by DCA and to guarantee that we can globally solve MILP, we combine DCA with a suitable Branch-and-Bound scheme (GOA-DCA). Some numerical results to the applications of DCA and GOA-DCA for (1) are also reported in the final section of the paper.

2 DC Reformulation for MILP

2.1 Reformulation of Integer Set

In this section, we will reformulate the integer set $\{y : y \in \mathbb{Z}_+^m\}$ using a twice continuously differentiable function.

Let

$$p(y) := \sum_{i=1}^{m} (1 - \cos(2\pi y_i)), \tag{2}$$

where $y \in \mathbb{R}^m$. The function $p : \mathbb{R}^m \to \mathbb{R}$ has some interesting properties which can be verified without any difficulty:

1) $0 \leq p(y) \leq 2m \ \forall y \in \mathbb{R}^m$;
2) $p(y) = 0$ if and only if $y_i \in \mathbb{Z}$, for all $i = 1, \ldots, m$;
3) $p(y) \in \mathbf{C}^\infty(\mathbb{R}^m, \mathbb{R})$.

By virtue of the third property, we can calculate the first and the second derivatives of $p(y)$, the gradient and the Hessian matrix, as follows

$$\nabla_y p(y) = 2\pi \begin{bmatrix} \sin 2\pi y_1 \\ \cdots \\ \sin 2\pi y_m \end{bmatrix} = 2\pi \sin 2\pi y, \tag{3}$$

$$\nabla_y^2 p(y) = 4\pi^2 \mathrm{Diag}(\cos 2\pi \mathrm{y}_1, \cdots, \cos 2\pi \mathrm{y}_\mathrm{m}), \tag{4}$$

where $\mathrm{Diag}(\cos 2\pi y_1, \cdots, \cos 2\pi y_m)$ is a $m \times m$ diagonal matrix.

Therefore, the spectral radius of $\nabla_y^2 p(y)$, denoted by $\rho(\nabla_y^2 p(y))$, satisfies the following inequality:

$$\rho(\nabla_y^2 p(y)) \ := \ \max_{1 \le i \le m} (4\pi^2 |\cos 2\pi y_i|) \le 4\pi^2. \tag{5}$$

With the help of the properties 1) and 2), the set $\{y : y \in \mathbb{Z}_+^p\}$ can be represented as

$$\{y : y \in \mathbb{Z}_+^p\} \ \equiv \ \{y : p(y) = 0, y \in \mathbb{R}_+^p\} \equiv \ \{y : p(y) \le 0, y \in \mathbb{R}_+^p\}. \tag{6}$$

2.2 Reformulation of MILP as a DC Program

Let $\mathbf{K} := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}_+^m : Ax + Gy \le b, A_{eq}x + B_{eq}y = b_{eq}, lb_x \le x \le ub_x, lb_y \le y \le ub_y\}$ be a nonempty and compact set.

The problem MILP (1) can be expressed as

$$\min\{f(x, y) := c^T x + d^T y : (x, y) \in \mathbf{K}, y \in \mathbb{Z}_+^m\}. \tag{7}$$

Using (6), we reformulate the problem (7) as an equivalent problem:

$$\min\{f(x, y) : (x, y) \in \mathbf{K}, p(y) \le 0\}. \tag{8}$$

We establish a penalized problem for (8) with a penalty parameter t (a positive constant):

$$\min\{F_t(x, y) := c^T x + d^T y + tp(y) : (x, y) \in \mathbf{K}\}. \tag{9}$$

Note that (9) is a nonlinear and nonconvex optimization problem. The additional item $tp(y)$ in the objective function satisfies the inequality $tp(y) \ge 0$ for all $y \in \mathbb{R}^m$, and it is equal to 0 if and only if $y \in \mathbb{Z}^m$. According to the general result of the penalty method (see [8], pp. 366-380), for a given large number t, the minimizer of (9) should be found in a region where p is relatively small.

Definition 1. (See [6].) Let $\tilde{y} \in \mathbb{Z}^m$. The set $\mathrm{N}(\tilde{\mathrm{y}}) = \{\mathrm{y} : ||\mathrm{y} - \tilde{\mathrm{y}}||_\infty \le \frac{1}{5}\}$ is called a $\frac{1}{5}$ - *cubic neighborhood* of the integer point $\tilde{y}$.

Theorem 1. *(See [6].)Suppose that t is large enough, if (x^*, y^*) is a global minimizer of (9) and y^* is in a $\frac{1}{5}$ - cubic neighborhood of an integer point $\tilde{y}$, then $(x^*, \tilde{y})$ is a solution of the problem (7).*

However, the problem (9) is a difficult nonlinear optimization problem. Fortunately, we can represent p as a DC function:

$$p(y) = (\frac{\eta}{2} y^T y) - (\frac{\eta}{2} y^T y - p(y)), \tag{10}$$

where $\eta \geq \rho(\nabla_y^2 p(y))$. Note that $(\frac{\eta}{2} y^T y) - (\frac{\eta}{2} y^T y - p(y))$ is a DC function (difference of two convex functions) if $\eta \geq \rho(\nabla_y^2 p(y))$. The reason is that the functions $\frac{\eta}{2} y^T y$ and $\frac{\eta}{2} y^T y - p(y)$ are convex, because their Hessian matrices are semi-positive definite matrices when the inequality $\eta \geq \rho(\nabla_y^2 p(y))$ satisfies. Using (5) in the section 2.1, $\rho(\nabla_y^2 p(y)) \leq 4\pi^2$, we can take $\eta = 4\pi^2$ to establish an available DC decomposition of p :

$$p(y) = 2\pi^2 y^T y - (2\pi^2 y^T y - p(y)).$$

Thus, the problem (9) can be reformulated as a DC programming problem:

$$\min\{F_t(x, y) := g(y) - h(x, y) : (x, y) \in \mathbf{K}\}. \tag{11}$$

where $g(y) := 2t\pi^2 y^T y$ is a convex quadratic function, $h(x, y) := (2t\pi^2 y^T y - tp(y) - d^T y) - c^T x$ is a separable convex function.

3 DCA for Solving Problem (11)

DC Algorithm (DCA) has been introduced by Pham Dinh Tao in 1985 as an extension of the subgradient algorithm, and extensively developed by Le Thi Hoai An and Pham Dinh Tao since 1993 to solve DC programs. It is actually one of the rare algorithms for nonlinear nonconvex nonsmooth programming which allows solving very efficiently large-scale DC programs. DCA has successfully been applied in solving real world nonconvex programs to which it quite often gives global solutions and is proved to be more robust and more efficient than the related standard methods, especially for large-scale problems. For more details of DC programs and DCA, the reader is referred to [1,2,3,4] and the references therein.

According to the general framework of DCA, we need constructing two sequences $\{X^k\}$ and $\{Y^k\}$. In our problem, $\{X^k := (x^k, y^k)\}$ and $\{Y^k := (u^k, v^k)\}$. In order to compute $Y^k = (u^k, v^k)$, we need computing subdifferential of the function h at the point $X^k = (x^k, y^k)$, denoted by $\partial h(x^k, y^k)$.

Definition 2. (See [1,2].) Let $\Gamma_0(\mathbb{R}^n)$ denote the convex cone of all lower semi-continuous proper convex functions on $\mathbb{R}^n$. For all $\theta \in \Gamma_0(\mathbb{R}^n)$ and $x_0 \in dom(\theta) := \{x \in \mathbb{R}^n : \theta(x) < +\infty\}$, $\partial\theta(x_0)$ denotes the subdifferential of θ at x_0

$$\partial\theta(x_0) := \{y \in \mathbb{R}^n : \theta(x) \geq \theta(x_0) + < x - x_0, y >, \forall x \in \mathbb{R}^n\}.$$

It is well-known that if θ is differentiable at x_0, then $\partial\theta(x_0)$ reduces to a singleton which is exactly $\{\nabla\theta(x_0)\}$.

In our problem, the convex function h was defined as $h(x, y) := (2\pi^2 t y^T y - tp(y) - d^T y) - c^T x$ which is a twice continuously differentiable function. Thus, $\partial h(x^k, y^k) = \{\nabla_{(x,y)} h(x^k, y^k)\}$. The vector $Y^k = (u^k, v^k)$ can be computed explicitly using the following equivalence:

$$(u^k, v^k) \in \partial h(x^k, y^k) \Leftrightarrow (u^k = -c, v^k = 4\pi^2 t y^k - 2\pi t \sin 2\pi y^k - d). \tag{12}$$

Let $g^*(y) := \sup\{< x, y > -g(x) : x \in \mathbb{R}^n\}$ be the conjugate function of g. For computing $X^{k+1} = (x^{k+1}, y^{k+1}) \in \partial g^*(u^k, v^k)$, we have to solve the convex quadratic program:

$$\min\{2\pi^2 t y^T y - \langle (x, y), (u^k, v^k) \rangle : (x, y) \in \mathbf{K}\}. \tag{13}$$

Every optimal solution of the problem (13) gives us one vector $X^{k+1} = (x^{k+1}, y^{k+1})$. Repeating the above operation, we can establish the sequences $\{X^k\}$ and $\{Y^k\}$.

DC Algorithm (DCA)

Initialization:
Choose an initial point $X^0 = (x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m$.
Let t be a large enough positive number.
Let ϵ_1, ϵ_2 be sufficiently small positive numbers.
Iteration number $k = 0$.

Repeat:
- Calculate $(u^k, v^k) \in \partial h(x^k, y^k)$ via (12).
- Solve the quadratic convex program (13) to obtain (x^{k+1}, y^{k+1}).
- $k \leftarrow k + 1$.

Until:
If either $||X^k - X^{k-1}|| \leq \epsilon_1(1 + ||X^{k-1}||)$
or $|F_t(X^k) - F_t(X^{k-1})| \leq \epsilon_2(1 + |F_t(X^{k-1})|)$
Then **STOP** and verify:
If y^k is in a $\frac{1}{5}$ - cubic neighborhood of an integer point $\tilde{y} \in \mathbb{Z}_+^m$ and $(x^k, \tilde{y})$ is a feasible solution to the problem (1)
Then $(x^k, \tilde{y})$ is a feasible computed solution
Else (x^k, y^k) is the computed solution.

The convergence of DCA can be summarized in the next theorem whose proof is essentially based on the convergence theorem of the general scheme of DCA (see [1,2,3]).

Theorem 2 (Convergence properties of DC Algorithm)

1. *DCA generates a sequence $\{(x^k, y^k)\}$ such that the sequence $\{F_t(x^k, y^k)\}$ is decreasing and bounded below.*
2. *If the optimal value of (11) is finite and the infinite sequences $\{X^k\}$ and $\{Y^k\}$ are bounded, then every limit point X^∞ of the sequence $\{X^k\}$ is a Karush-Kuhn-Tucker point.*

4 A Combination of DCA with a B&B Scheme

In order to evaluate the quality of the solution obtained by DCA and improve the computed solution of DCA for finding a global optimal solution, we propose a

hybrid method which combines DCA with an adapted Branch-and-Bound scheme for globally solving MILP.

Branching. Suppose that we have already found a computed solution by DCA, denoted by (x^*, y^*). If it is not feasible solution of MILP, then we can find an element $y_i^* \notin \mathbb{Z}_+$, and the subdivision is performed in the way that $y_i \leq \lfloor y_i^* \rfloor$ or $y_i \geq \lceil y_i^* \rceil$(where $\lfloor y_i^* \rfloor$ (resp. $\lceil y_i^* \rceil$) means the floor (resp. ceil) number of y_i^*). The two subdivision problems can be described as

$$\min\{F_t(x,y) = g(y) - h(x,y) : (x,y) \in \mathbf{K}, y_i \leq \lfloor y_i^* \rfloor\}. \tag{14}$$

$$\min\{F_t(x,y) = g(y) - h(x,y) : (x,y) \in \mathbf{K}, y_i \geq \lceil y_i^* \rceil\}. \tag{15}$$

Note that (14) and (15) are also DC programs. The feasible sets of these problems are disjunctive, and the union of the two feasible sets might be smaller than $\mathbf{K}$, but it includes every feasible solution of the original MILP problem (1).

Bounding. Solving directly the subproblems (14) and (15) is a difficult task and there is no extraordinary efficient and practical solution method. So we establish their lower bound problems:

$$\min\{f(x,y) = c^T x + d^T y : (x,y) \in \mathbf{K}, y_i \leq \lfloor y_i^* \rfloor\}. \tag{16}$$

$$\min\{f(x,y) = c^T x + d^T y : (x,y) \in \mathbf{K}, y_i \geq \lceil y_i^* \rceil\}. \tag{17}$$

Note that, the problem (16) (resp. (17)) is a lower bound problem to (14) (resp. (15)), because $F_t(x,y) = c^T x + d^T y + tp(y) \geq c^T x + d^T y$ for all $(x,y) \in \mathbf{K}$. It is clear that (16) and (17) are linear programs, so there are efficient practical algorithms for solving them even if they are large-scales. After solving a lower bound problem, the best current upper bound solution will be updated if a better feasible solution was discovered.

DCA + B&B for Solving MILP. The B&B method forms a search tree in a way that we always create branches with a node who has a smallest lower bound (a node here means a subdivision problem). During the search process, we can prune away every node who is an infeasible problem or whose lower bound is greater than the best current upper bound.

The algorithm terminates when every node of the search tree was pruned away or the gap between the best upper bound and the current minimal lower bound is less than a given tolerance.

When DCA is restarted?
A suggestion to restart DCA in some steps of B&B helps to reduce the computational time. We have several heuristic strategies to restart DCA:

- *The first strategy*: When a best current upper bound solution is discovered, we can restart DCA from this upper bound solution. Perhaps this strategy will find a better feasible solution. If a better feasible solution was found by DCA, the best current upper bound solution will be updated.

– *The second strategy*: When we choose a current minimal lower bound node to create branches, we can restart DCA from this best lower bound solution. Perhaps this strategy will find a better feasible solution for updating the best current upper bound.

Based on the above discussions, we have established a Global Optimization Algorithm, denoted by GOA-DCA, for solving general MILP. Because the length of this article is limited, for more details about GOA-DCA, the reader is referred to the technical report [9].

Note that, using this solution method for solving ILP, we just need to ignore the coefficients of the continuous variables, i.e. the vectors c, lb_x, ub_x and the matrices A, A_{eq} in the problem (1).

5 Computational Experiments

We have implemented the algorithm DCA and GOA-DCA in MATLAB R2007a and tested it on a laptop equipped with Windows XP Professional, Genuine Intel(R) CPU T2130 1.86GHz, 2G RAM. The version of C++ has also been implemented using CPLEX and COIN-OR.

In this section, we present an illustrative example of an integer linear program. More tests on MILP and ILP (medium-scales or large-scales) have been realized with a series of data from MIPLIB 3.0 [10], they are real world pure or mixed integer linear problems, some of them are large-scales ($10^5 - 10^6$ integer variables) and difficult to be globally solved.

An illustrative example to an integer linear program:
This integer linear program is defined as

$$\min\{d^T y : By \leq b, lb_y \leq y \leq ub_y, y \in \mathbb{Z}_+^m\}. \tag{18}$$

– Number of integer variables: $m = 12$
– Number of linear inequality constraints: 7

$d = -(96, 76, 56, 11, 86, 10, 66, 86, 83, 12, 9, 81)^T$

B =

19	1	10	1	1	14	152	11	1	1	1	1
0	4	53	0	0	80	0	4	5	0	0	0
4	660	3	0	30	0	3	0	4	90	0	0
7	0	18	6	770	330	7	0	0	6	0	0
0	20	0	4	52	3	0	0	0	5	4	0
0	0	40	70	4	63	0	0	60	0	4	0
0	32	0	0	0	5	0	3	0	660	0	9

$b = (18209, 7692, 1333, 924, 26638, 61188, 13360)^T$
$lb_y = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$
$ub_y = (26638, 26638, 26638, 26638, 26638, 26638, 26638, 26638, 26638, 26638, 26638, 26638)^T$

Using the Branch-and-Bound method for solving (18):

- The global optimal solution was found after 182 iterations.
- The global optimal value: -261922.
- The global optimal solution: [0,0,0,154,0,0,0,913,333,0,6499,1180].
- The average CPU time: 13.12 seconds.

Using DCA for solving (18):
We have tested the performance of DCA with different starting points. Here are some numerical results in Table 1:

Table 1. DCA test results for (18) with $\epsilon_1 = \epsilon_2 = 1e-3$ and $t = 1000$

TestNo	Starting point	#Iter	CPU time (secs.)	Obj (Min)	Integer solution (T/F)
1	$ub - lb$	30	0.5183	-261890	T
2	$\frac{ub-lb}{2}$	30	0.5079	-261890	T
3	$\frac{ub-lb}{3}$	30	0.4950	-261890	T
4	$\frac{ub-lb}{4}$	30	0.4913	-261890	T
5	$\frac{ub-lb}{5}$	30	0.5165	-261890	T
6	$\frac{ub-lb}{6}$	30	0.5306	-261890	T
7	$\frac{ub-lb}{8}$	30	0.5505	-261890	T
8	$\frac{ub-lb}{9}$	11	0.2583	-261657.341	F
9	$\frac{ub-lb}{20}$	21	0.3978	-249604.313	F
10	$\frac{ub-lb}{50}$	12	0.2396	-230200	T
11	$\frac{ub-lb}{100}$	10	0.2085	-137045	T

Table 1 shows:

- DCA often gives a feasible solution (TestNo 1-7, 10 and 11).
- Using different staring points, DCA often converges with a small number of iterations (less than 30 iterations).
- Most of the objective values in Table 1 are the number -261890. This number is close to the global optimal value which is -261922.

We calculate the relative error by the formula:

$$err := \frac{|\text{optimal value obtained by DCA} - \text{global optimal value}|}{|\text{global optimal value}|}.$$

We get $err = |-261890 + 261922|/261922 \approx 1.2E-4$. Such a small error shows the computed solutions of DCA are often close to the global optimal solution.

More tests for large-scale problems show that the good performance of DCA is not particular to this illustrative example, but a universal result. Especially, the iteration numbers are often relatively small (less than 60 for the tests with $10^5 - 10^6$ variables).

Using GOA-DCA for solving (18):
Here are some test results of GOA-DCA without restarting DCA during the B&B process:

– The optimal solution was discovered after 36 iterations.
– The optimal value: -261922.
– Theoptimalsolution:[0,0,0,154,0,0,0,913,333,0,6499,1180](globallyoptimized).
– The CPU time is 2.35 seconds $\ll$ 13.12 seconds (by B&B).

More test results show that GOA-DCA can always find the global optimal solution if it exists, and the CPU time is always smaller than B&B. The superiority of GOA-DCA with respect to B&B increases with the dimension. An interesting issue is how to restart DCA. More tests for large-scale problems show that the two strategies for restarting DCA play a quite important role in GOA-DCA method. DCA often finds rapidly a feasible solution and it improves considerably the best current upper bound during the B&B process and therefore accelerates the convergence of GOA-DCA. For more discussions of the tests on DCA and GOA-DCA, the reader is referred to the technical report [9].

6 Conclusion and Future Work

In this paper, we have presented a new continuous nonconvex optimization approach based on DC programming and DCA for solving the general MILP problem. Using a special penalized function, we get a DC program. With a suitable penalty parameter and a good starting point, DCA generates a sequence for which the objective values decrease and converge very fast to a computed solution (which is often a feasible solution of MILP and close to a global optimal solution). Despite its local character, DCA shows once again, its robustness, reliability, efficiency and globality. The hybrid method GOA-DCA which combines DCA with an adapted Branch-and-Bound aims at checking the globality of DCA and globally solving the MILP problem efficiently. Preliminary numerical tests show that GOA-DCA usually gives the global optimal solution much faster than B&B method.

This paper could be considered as a first step of using the DC Programming approach for solving general MILP and ILP problems. Some extensions of this solution method are in development. For instance, we are trying to integrate GOA-DCA with some cuts (Gomory's cut, lift-and-projet cut, knapsack cover, etc.). We also want to extend this approach to more difficult nonlinear MIPs and pure Integer Programs:

1. Pure Integer Nonlinear Program (convex objective function, and DC objective function).
2. Mixed Integer Quadratic Program (convex objective function, and nonconvex objective function).
3. Mixed Integer Nonlinear Program with DC objective function, etc.

Mixed Integer Quadratic Programs with binary integer variables have already been treated in another work [5]. Results concerning these extensions will be reported subsequently.

References

1. Pham Dinh, T., Le Thi, H.A.: DC Programming. Theory, Algorithms, Applications: The State of the Art. LMI, INSA - Rouen, France (2002)
2. Pham Dinh, T., Le Thi, H.A.: Convex analysis approach to D.C. programming: Theory, Algorithms and Applications. Acta Mathematica Vietnamica 22(1), 287–367 (1997)
3. Pham Dinh, T., Le Thi, H.A.: The DC programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems. Annals of Operations Research 133, 23–46 (2005)
4. Pham Dinh, T., Le Thi, H.A.: DC optimization algorithms for solving the trust region subproblem. SIAM J. Optimization 8, 476–507 (1998)
5. Niu, Y.S.: Programmation DC et DCA pour la gestion du portefeuille de risque de chute du cours sous des contraintes de transaction. LMI, National Institute for Applied Sciences - Rouen, France (2006)
6. Ge, R.P., Huang, C.B.: A Continuous Approach to Nonlinear Integer Programming. Applied Mathematics and Computation 34, 39–60 (1989)
7. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience Publication, Chichester (1999)
8. Luenberger, D.G.: Linear and Nonlinear Programming, 2nd edn. Springer, Heidelberg (2003)
9. Pham Dinh, T., Niu, Y.S.: DC Programming for Mixed-Integer Program. Technical Report, LMI INSA-Rouen (2008)
10. MIPLIB 3.0, http://miplib.zib.de/miplib3/miplib.html
11. COIN-OR, http://www.coin-or.org/

Simulation-Based Optimization for Steel Stacking

Rui Jorge Rei[1], Mikio Kubo[2], and João Pedro Pedroso[3]

[1] Faculdade de Ciências da Universidade do Porto, Portugal
rui.rei@alunos.dcc.fc.up.pt
[2] Tokyo University of Marine Science and Technology, Japan
kubo@kaiyodai.ac.jp
[3] Faculdade de Ciências da Universidade do Porto and INESC - Porto, Portugal
jpp@fc.up.pt

Abstract. In many sectors of industry, manufacturers possess warehouses where finished goods are stored, awaiting to fulfill a client order.

We present a situation where these items are characterized by release and due dates, i.e. warehouse arrival for storage and client delivery, respectively. The warehouse has a number of positions available, where items can be placed on top of each other, forming stacks. For item manipulation, there is a single a stacking crane, able to carry one item at a time.

When in a given stack an item at the top is due at a date later than some item below it, it must be relocated to another stack, so that the item below can be delivered. In this problem the objective is to minimize the number of movements made by the crane.

1 Introduction

We present an applied stacking problem arising in steel manufacturing, which we name as *steel stacking problem*. This combinatorial optimization problem has many variants, with applications e.g. in container stacking [2,3] and ship stowage planning [1]. Here we will focus on a variant where items are produced at given dates (the release dates), and are then placed in the warehouse, where they remain until being shipped to the customers, on dates that are also predetermined (the due dates).

The warehouse is divided into several positions, each having its own stack for placing items. There are no particular restrictions on item placement.

At the time of item retrieval for delivery, it may be required to relocate items on top of the departing item, before actually delivering it. These relocations are called *reshuffling* or *shifting*, meaning unproductive moves of an item, which are required to access another item that is stored beneath it. In our case, reshuffling occurs when an item is placed above another that is due first, which we call an *inversion*.

Each of these movements is made by a crane, which is able to move one item at a time. As the operation of the crane is energy and time consuming, the objective

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 254–263, 2008.

is to minimize the number of movements it does. This problem is *NP-hard* [1], a fact which makes finding optimal solutions inviable in polynomial time, with known algorithms.

We propose a solution method which consists of using a custom discrete event simulation engine, based on a heuristics for deciding the placement for each item. The heuristics is semi-greedy, meaning that it has a random component besides the greedy behavior. Since simulation is cheap when compared to "classic" resolution methods, such as branch-and-bound, or local search, we are allowed to run several simulations, leading to a set of different solutions, from which the best is selected at the end.

The paper is structured as follows. First, we take a short literature overview of scientific work on stacking problems. Subsequently, we present a detailed problem description. A solution is then proposed, along with three simple stacking strategies that were compared on 1500 problem instances. Finally, the results and some ideas for future work are discussed.

1.1 Literature Overview

Dekker et al. [2] present a simulation study of container terminal activities, using category and residence time stacking policies. Also, they mention the existence of little published work on stacking problems. Since this problem has less restrictions than those of container stacking, the stacking policies there described are unsuitable in our simplified problem.

Though the problem handled in this paper is a rather simple stacking variant, with very few restrictions, it is nonetheless *NP-hard* (*NP-completeness* has been shown in [1] for a similar problem, arising in ship stowage planning).

1.2 Contributions

Our contributions are the following: we formalize the description of a stacking problem and propose several heuristics for constructing solutions in a semi-greedy way. We define a set of benchmark instances for this problem, and compare the results obtained by several heuristics on the semi-greedy construction.

2 Formal Description

Consider the problem of stacking n items in a warehouse with p positions. We denote items as $i_1, i_2, \ldots, i_n$, and stacks as $s_1, s_2, \ldots, s_p$.

Let $r \in \mathbb{N}^n$ be the list of release dates, where r_i denotes the release date of item i. In the same way, let $d \in \mathbb{N}^n$ be the list of due dates. We assume that due dates are greater than release dates for every item, i.e. $r_i < d_i$, for $i = 1, \ldots, n$.

The objective is to store all the items, and deliver them to the client, with the minimum number of crane movements. The solution is represented as an ordered list of movements, where we represent the movement of an item from stack s_o to stack s_d as $s_o \rightarrow s_d$ (note that only the item on the top of the stack is moved). The time at which a movement occurs is either the instant of release of an item

(when the movement corresponds to storing it in the warehouse), or the instant of delivery of an item (for the delivery movement itself, and all the underlying reshuffling moves).

The order of moves in the movement list must always respect release and due dates for the solution to be valid. Furthermore, there cannot exist movements originating at empty stacks. Regarding intermediate moves, they can be in any number, and between any two positions, as long as they occur between the release and due dates of the associated item.

3 Proposed Solution

For simulating systems where changes are known to occur only at certain points in time, the intervals between system changes can be safely ignored. This is the basis of discrete event simulation [5,4], a well-known simulation discipline. Since we know *a priori* the discrete times when arrivals and departures will take place, using discrete event simulation seems appropriate, and far less expensive than a complete search algorithm such as branch-and-bound. Other alternatives, as the local search or tabu search meta-heuristics, are inappropriate, since any change of a part of the solution implies very deep changes in the remaining part, rendering it virtually useless.

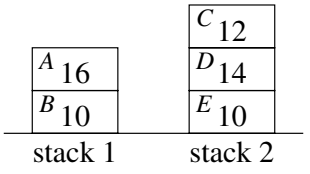
We will now describe how events are handled, highlighting when placement heuristics are used, and describe the event preprocessing performed by the simulator during the setup phase. The setup phase takes place before the beginning of the simulation, when the instance data is used by the simulator to build a chronologically ordered event schedule.

3.1 Simulation Setup

We have built a custom discrete event simulation engine, with two event types:

Item release: when an item is released, its location in the warehouse is determined through a call to a placement heuristics. The heuristics scores stacks in the warehouse, building a list of best scored candidates. After the list is created, a stack s is randomly chosen among the best candidates and the item is placed in s. Solution diversity is granted by this random target choice. If two or more items are to be released simultaneously, the corresponding release events are processed in the inverse order of their due dates, thus avoiding inversions among items arriving together.

Item delivery: at simulation time t all items with due date $d_i = t$ must be delivered. All items at the top of a stack with delivery time t are immediately delivered. For all the remaining items with $d_i = t$ that were not delivered in the previous step, the simulation engine searches the item's stack, and swaps all items above it to other stacks, finally delivering the target items. Items are delivered in the inverse order of their depth in the corresponding stack, as illustrated in Figure 1. The position for the items begin reshuffled



Unordered delivery:

1. $\{C, D\} : s_2 \rightarrow s_1$
2. $E : s_2 \rightarrow$ client
3. $\{D, C, A\} : s_1 \rightarrow s_2$
4. $B : s_1 \rightarrow$ client

Depth-ordered delivery:

1. $A : s_1 \rightarrow s_2$
2. $B : s_1 \rightarrow$ client
3. $\{A, C, D\} : s_2 \rightarrow s_1$
4. $E : s_2 \rightarrow$ client

Fig. 1. Unordered delivery (center) and depth-ordered delivery (right) on a two-stack warehouse (left), where two items, B and E, are due at $t = 10$. If no particular order is used, E may be delivered before B, requiring 7 moves to deliver the two items. However, if shallower items are delivered first (i.e., B is delivered before E, as shown at the right), only 6 moves are required.

is determined by the placement heuristics used at item release, this time not allowing the original stack to be a candidate target.

If at any time release and delivery events occur simultaneously, the deliveries are handled first (in discrete event simulation terminology, this corresponds to assigning a higher scheduling priority to deliveries).

4 Stacking Strategies

In this section, we propose several stacking strategies for item placement, both when items are arriving at the warehouse after being released, or when reshuffling a stack in order to reach an item for delivery.

4.1 Minimize Conflicts

This is the simplest heuristics presented, and it is the starting point of our research. The aim of this heuristics is to avoid placing an item i, with due date d_i, above any item j with due date $d_j < d_i$. Defining $E(s)$ as the earliest due date for items in stack s,

$$E(s) = \min\{d_i : i \in s\}$$

we say that a conflict, or inversion, is created if we place item j on stack s, and $d_j > E(s)$. Let us also define $I(s)$ as the number of inversions in stack s:

$$I(s) = |\{i \in s : d_i > E(s) \text{ and } i \text{ is above the earliest departing item}\}|$$

Figure 2a presents a simple example, for a two-stack warehouse. With a conflict minimization strategy, item F (the arriving item) will be placed in stack s_1, since placing it in s_2 will cause a new inversion (as $d_F > E(s_2)$).

Note that although s_1 has an inversion, it is the chosen stack for item F, since we are only considering the creation of new inversions. Even though there is a zero-conflict stack (s_2), it is not a good choice, since we would be creating an inversion after placing F on top of C.

4.2 Delay Conflicts

Delaying conflicts is a heuristics that derives directly from the previous, in the sense that its aim is also to avoid creating inversions.

The only difference is that, when placing an item i such that for any stack s, $d_i > E(s)$ (i.e. when i creates a new inversion no matter which stack it goes to), we prefer the stack with highest $E(s)$. This makes the newly created inversion express later in time, as illustrated in Figure 2b.

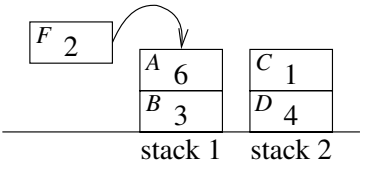


Fig.2a: Conflict minimization heuristics. There are two stacks, with earliest due dates 3 and 1. The item arriving has a delivery date of 2, and hence it will cause a conflict in stack 2 but not in stack 1.

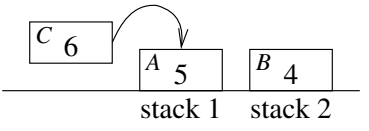


Fig.2b: Conflict delaying heuristics. There are two stacks, with earliest due dates 5 and 4; the arriving item has a delivery date of 6, and hence it will cause a new conflict in both stacks. Stack 1 is chosen, as there the conflict will show up later.

Fig. 2. Conflict minimization (left) and conflict delaying (right) heuristics

Conflict delaying is done hoping that when the simulation clock reaches $E(s)$, a different stack s' will be able to receive i without conflicts. Intuitively, this seems to be a good strategy when space utilization becomes low, i.e., if stacks with no inversions are likely to be found later.

Figure 3 shows an example of the advantage of delaying conflicts in a situation where a new inversion is mandatory. Since $E(s_1) < d_C$ and $E(s_2) < d_C$, a new inversion is unavoidable. The conflict minimization heuristics sees both stacks as equal, and randomly selects its target. Let us consider the case where s_2 is selected. Figure 3 (center) shows the movement list in this case, where 6 moves are necessary to deliver all items. There are two unproductive moves of C.

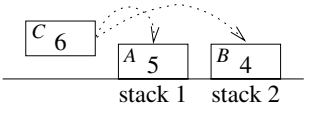
Now consider delaying the conflict (right). Since $E(s_1) > E(s_2)$, stack s_1 is chosen for item C. In this case, only 5 movements are required for delivering all items.

We have gained a move because the conditions in the warehouse changed between the release of item C (at time $r_C < 4$) and the earliest item delivery from stack s_1, (which occurs at time $E(s_1) = 5$), allowing us to reshuffle item C to s_2 (move 3 in the right list) when it was already empty.

4.3 Flexibility Optimization

We define flexibility for a stack s as the size of the maximum set of items with different due dates that can be placed on s without causing new inversions. Let us call this set $\mathcal{F}(s)$, and define flexibility as

$$F(s) = |\mathcal{F}(s)|.$$



Random choice:

1. C : release $\rightarrow s_2$
2. $C : s_2 \rightarrow s_1$
3. $B : s_2 \rightarrow$ client
4. $C : s_1 \rightarrow s_2$
5. $A : s_1 \rightarrow$ client
6. $C : s_2 \rightarrow$ client

Delayed conflict:

1. C : release $\rightarrow s_1$
2. $B : s_2 \rightarrow$ client
3. $C : s_1 \rightarrow s_2$
4. $A : s_1 \rightarrow$ client
5. $C : s_2 \rightarrow$ client

Fig. 3. Random stack choice (center) and delayed conflict stack choice (right) for a case with a mandatory inversion (left)

The greater $F(s)$, the more items the stack can receive without conflicts. Flexibility can also be seen as $E(s)$, since we can only put a set with $E(s)$ items of different, decreasing due dates, on top of s without causing inversions. This idea is illustrated in figure 4, where it is clear that the number of items the stack "accepts" is equal to its earliest due date.

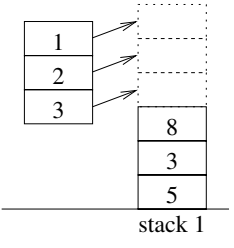


Fig. 4. Example of stack flexibility, with $\mathcal{F}(s)$ shown in the left. The numbers inside the boxes represent the items' due dates, d_i.

By looking at stack flexibility we get a better judgement of stacks where no inversions are caused by an arriving item. Figure 5 shows a two-stack example of an arrival where the item does not cause a conflict with any of the two stacks. Both the conflict minimization and conflict delaying strategies would see the two stacks as equal candidates.

The flexibility variation caused by placing item i on stack s,

$$\Delta F(i, s) = d_i - E(s),$$

provides a different view of the two stacks, and it becomes clear that we lose more flexibility by choosing stack s_1 (with $\Delta F(C, s_1) = -3$) rather than stack s_2 (with $\Delta F(C, s_2) = -1$).

Note that this does not represent an immediate gain, but nevertheless maintaining high flexibility is a good idea for preparing future arrivals. Suppose that,

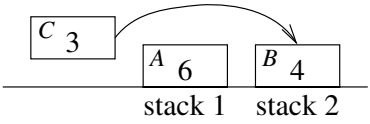


Fig. 5. Flexibility based decision in a multiple no-inversion case

after C, an item D arrives, with due date $d_D = 5$. If we had selected s_1 for C, we would now be at hand with an unavoidable inversion, or a preparation move would be required ($C : s_1 \rightarrow s_2$). This situation is avoided by selecting s_2 in the first place.

This means that it is desirable to maximize $\Delta F(i, s)$ in cases where no-inversion stacks exist. On the other hand, when a new inversion is mandatory, we have seen cases where delaying the conflict may be advantageous. In order to capture these two cases, we divide our goal in two parts:

1. if there is a stack s such that $\Delta F(i, s) \leq 0$, then we are not causing a new inversion, and we want to maximize flexibility.
2. if there is no stack s such that $\Delta F(i, s) \leq 0$, a new inversion is unavoidable, and we want the minimum value of flexibility variation. Note that this is equivalent to delaying the conflict.

To summarize, in this heuristics the aim is trying to approach $\Delta F(i, s)$ to 0, but always preferring a negative variation to a positive variation (as the latter represents new inversions).

5 Computational Results

For testing the heuristics we have prepared a set of 1500 different instances, with varying sizes (number of stacks and number of items), and different values for parameters controlling the sparsity of release and due dates, as well as the residence time of items.

The instances are divided into two main classes: easy and hard. The major difference between them is the relation between warehouse size and item count. Hard instances have a small number of stacks for a relatively large number of items, while easy instances have a medium/high number of stacks for a proportionally smaller set of items.

For every instance, each heuristics was used on 100 simulations, each simulation corresponding to a different seed in the random number generator (RNG). We collected the best result of each heuristics among these 100 simulations for each instance.

In the results we present the move ratio for all heuristics, for all the instance sizes and categories.

The move ratio is defined as the number of moves of the solution divided by a known lower bound of the move count. The lower bound considered here is

Heuristics	Instances sizes					
	$p = 10, n = 100$		$p = 20, n = 400$		$p = 40, n = 1600$	
	mean	variance	mean	variance	mean	variance
Minimize conflicts	1.123	0.01324	1.629	0.2268	3.036	2.192
Delay conflicts	1.075	0.004072	1.218	0.02823	1.492	0.1369
Optimize flexibility	1.007	0.0001978	1.027	0.0007952	1.080	0.004220

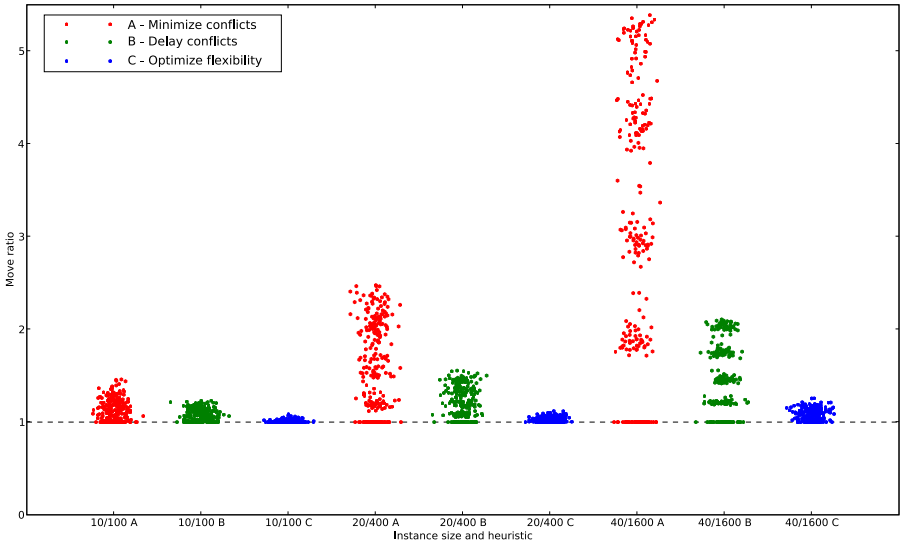


Fig. 6. Move ratio results for the 750 *easy* instances. (The points in the graphic are shifted in the x-axis by a random guassian value to improve visibility.)

twice the number of items in the instance: one move for arrival, and another for departure, corresponding to the ideal situation of having no reshuffling.

We determined the move ratio using the best result from the 100 independent simulations, for each instance-heuristics pair.

Figure 6 shows the results for the three heuristics in 750 "easy" instances, with 10, 20 and 40 stacks, and, respectively, 100, 400 and 1600 items. The average and variance of the move ratio are shown in the table coupled to the figure. In the same way, figure 7 shows the results for the three heuristics in 750 "hard" instances, with 2, 3, and 4 stacks, and, respectively, 32, 243 and 1024 items. For hard instances there are fewer stacks available, and hence there is less suppleness when placing items.

The results obtained in this section allow us to conclude that flexibility optimization is better than the other two heuristics in all contexts (varying size and problem difficulty).

Heuristics	Instances sizes					
	$p = 2, n = 32$		$p = 3, n = 243$		$p = 4, n = 1024$	
	mean	variance	mean	variance	mean	variance
Minimize conflicts	1.396	0.08011	3.984	2.393	12.96	43.65
Delay conflicts	1.357	0.07056	3.038	1.172	7.413	13.73
Optimize flexibility	1.368	0.09244	2.589	0.6888	5.391	5.977

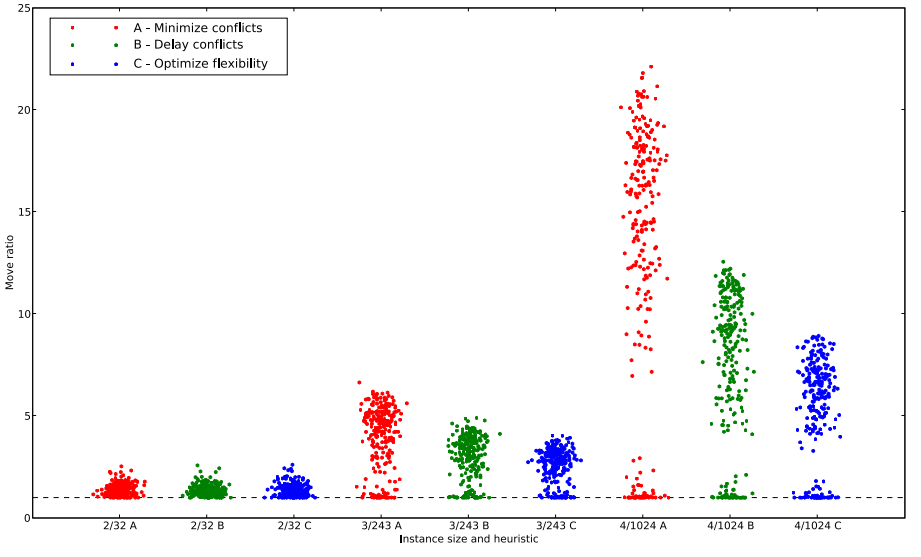


Fig. 7. Move ratio results for 750 *hard* instances. (The points in the graphic are shifted in the x-axis by a random guassian value to improve visibility.)

6 Conclusions and Future Work

In this work we tackle the problem of minimizing the number of movements made by a crane when storing items, characterized by a release date and a due date, in a warehouse with several stacks.

The solution strategy consists of using a custom discrete event simulation engine, based on a heuristics for deciding the placement for each item. We proposed three different semi-greedy heuristics, each having a random component besides a greedy behavior. Since simulation is computationally inexpensive, each heuristics is run on several independent simulations, leading to a set of different solutions, from which the best is selected. This provides a practical solution strategy to a problem which is rather difficult to formulate, let alone solve exactly.

The results obtained in preliminary tests show a very good performance, especially for the flexibility optimization heuristics. This conclusion was drawn from a statistically meaningful set of data, indicating that heuristics as the choice for industry implementation.

Future research directions include the implementation of a limited depth branch-and-bound algorithm to produce more accurate stack scoring values (because of the provided lookahead), and of remarshalling heuristics, that perform preparation moves before the arrival of an item that will cause a mandatory new inversion.

References

1. Avriel, M., Penn, M., Shpirer, N.: Container ship stowage problem: complexity and connection to the coloring of circle graphs. Discrete Appl. Math. 103(1-3), 271–279 (2000)
2. Dekker, R., Voogd, P., van Asperen, E.: A general framework for scheduling equipment and manpower at container terminals. OR Spectrum (2006)
3. Hartmann, S.: A general framework for scheduling equipment and manpower at container terminals. OR Spectrum (2004)
4. Pidd, M.: Computer simulation in management science, 4th edn. Wiley, Chichester (1988)
5. Robinson, S.: Simulation - The practice of model development and use. Wiley, Chichester (2004)

Robust Hedging of Electricity Retail Portfolios with CVaR Constraints

Marina Resta[1,⋆] and Stefano Santini[2]

[1] DI.E.M., sez. Matematica Finanziaria,
via Vivaldi 5, 16126 Genova, Italy
mresta@unige.it
[2] Enel SpA, Via Dalmazia 15, 00198, Roma, Italy
stefano.santini@enel.it

Abstract. This paper suggests a way for electricity retailers to build their supply portfolios, calibrating their exposure to physical and financial contracts, in order to hedge from risks that variously affect the supply side in power markets. In particular, we formulate an allocation model which describes uncertainty sources using the robust approach originally introduced by Soyster (1973), and we provide an explicit form to robust risk constraints. The notable elements of innovation of this paper include: (a) the focus on the optimization problem faced by retailers, which is generally less explored than its counterpart in the generation side; (b) the analysis of uncertainty sources through the robust optimization paradigm, and (c) the representation of robust constraints based on Conditional Value at Risk (CVaR).

Keywords: Conditional Value at Risk, Energy Management, Robust Optimization, Supply Side.

1 Introduction

As widely known, the introduction of competition and consumer choice (i.e. the deregulation) boomed on power markets all over the world in the past two decades. Such newly organised markets typically include one or more of the following structures:

– *Day–ahead (spot) Market.* This is the natural place where the bids (both from generation and supply side) are submitted. The market is cleared on the day before the actual dispatch. The day to be scheduled is divided into Nh periods; every bidding firm makes a price bid for each generation unit for the whole day. Commonly, in the day–ahead market either hourly contracts (for the 24 hours of the calendar day) or block contracts (i.e. a number of successive hours) are being traded.

⋆ Corresponding author.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 264–272, 2008.

– *Adjustment or Balancing Market.* The existence of this intra–day market (closing a few hours before delivery), is due to the long time span between the settling of contracts on the day–ahead market and physical delivery. It enables the participants to improve their balance of physical contracts in the short term.

This renewed scenario has introduced additional elements of complexity in the decision process of players, and a general shift in risk exposure of market participants: whereas at the beginning the risk was borne upon final electricity consumers through regulated tariffs, in the actual context the risk perception of different market stakeholders has changed.

Such remarks hold particularly on the retailers' side, where we are focusing on. It is a matter of fact that supply and demand bids are created on forecasts of customer usage; this means that if customer demand for power drops unexpectedly to a very low level, the retailer will be still asked to buy the total amount that the market bid requires. At this point, in order to avoid financial losses, the retailer will be forced to put on the balancing market the extra power not used by the customer. On the other hand, if customer demand surges, the supplier will be obliged to purchase extra power on the balancing market, to meet the excess of customer demand. In practice, both situations lead to major risk exposures, with the supplier facing volume risk (physical risk) and price risk (financial risk).

However, despite of consequences coming on balance sheets from improper allocation, building retailers portfolios has been only partially addressed in recent literature [7], [8], [9], [10]. Following this rationale, we are then going to suggest an approach to the problem based on the paradigm of robust optimization.

Robust optimization [12],[1],[3] concerns a way to manage uncertainty within optimization problems: whereas in the standard stochastic approach the optimal solution is obtained through a convenient probabilistic representation of problem parameters, the robust counterpart finds out solutions that are optimal over the finite (and convex) input space $\mathcal{A}$, for every parameter belonging to a proper deterministic uncertainty set $\mathcal{U}$. The method is actually applied to a variety of practical cases, with a certain preminent attention to financial applications (see, for instance: [2],[4]), as alternative to the classical optimization scheme *á la Markowitz* [11]. In all the cited cases, however, the way $\mathcal{U}$ is built is of fundamental importance for the success of the whole procedure; to such aim, our paper will follow the path traced by [5] and [6] through the representation theorem which is given below.

Theorem 1. *Let $\mathcal{Q}$ be a family of probability measures s.t.:*

$$\rho(X) = \sup_{Q \in \mathcal{Q}} \mathbb{E}_Q[X], \quad \forall X \in \mathcal{X} \ . \tag{1}$$

If the risk measure ρ is generated by $\mathcal{Q}$, then the following constraints are equivalent:

$$\rho(\tilde{\mathbf{r}}'\mathbf{x} - \mathbf{b}) \leq 0 \Leftrightarrow \tilde{\mathbf{r}}'\mathbf{x} \leq b, \quad \forall \mathbf{r} \in \mathcal{U}$$

$$\textit{where: } \mathcal{U} = conv\left(\left\{\sum_{i=1}^{N} q_i \mathbf{r}_i : \mathbf{q} \in \mathcal{Q}\right\}\right) \subseteq conv(\mathcal{A}) \ .$$

With respect to the existing literature, our contribution adds some elements of innovation that we are going to list. First of all, from the practical standpoint, we will focus on electricity markets on the retailers' side, and we will suggest an optimization scheme to build their portfolios, where both physical and financial constraints are considered. The allocation problem will be managed within a robust framework, in the sense outlined above. We will then show how robust (according to Eq.(1)) risk constraints may be built using Conditional Value at Risk (CVaR) as risk measure. In this way, in addition to the robustness of the optimal solution, we will also get a highly tractable optimization scheme, thanks to the linearity of the inserted CVaR constraints.

What remains of the paper is structured as follows. Section 2 will introduce the basic assumptions and variables on which our model relies. Section 3 will describe and discuss our proposed robust optimization scheme, to conclude with the representation of CVaR constraints in an explicit form. Section 4 will examine the results of the robust optimization procedure in a case study, and compare them to those of the fully deterministic solution. Finally, Section 5 will end the paper.

2 Basic Assumptions and Notational Conventions

We consider a simplified but meaningful structure of the regulatory framework, built up by picking relevant features from regulations of various European countries. We move into a fully liberalized retail market, where either business or domestic costumers can stipulate contracts with retailers for electricity supply, despite of the quantity of energy yearly consumed. The local distribution company will provide connection services between customers and distribution grid, including electricity metering, and technical maintenance. We then assume that the retailer's activity may be disassembled into four macro–components:

(a) the selling activity to customers; the retailer can stipulate a number NC of different selling contracts, diversified and standardised according to the type of customer, and to the time profile. As a rule, this activity should generate revenues:

$$R_{SELL} = \sum_{i=1}^{NC} \sum_{h=1}^{Nh} P_S(i,h)\ E(i,h)\ . \tag{2}$$

where $P_S(i,h)$ is the selling price for the $i-$th type of contract at hour h, and $E(i,h)$ is the corresponding amount of effectively consumed energy.

(b) The supplying activity; as a consequence of the selling activity, the retailer must enter the spot market to supply the estimated power load, acquiring electricity from Ns different sources. This implies a cost to be borne on the retailer, which may be expressed as:

$$C_{SP} = \sum_{s=1}^{Ns} \sum_{h=1}^{Nh} P_{SP}(s,h)\ B(s,h)\ . \tag{3}$$

where $P_{SP}(s,h)$ is the price for the $s-$th contract and $h-$th hourly profile on the spot market, and $B(s,h)$ is the related energy bandwidth, expressed in Mega Watt per hour (MWh).

(c) The balancing activity; since task (a) is based on effective demand for electricity, while (b) assumes a forecast of customers usage, a bias can arise between the amount of consumed energy, and the quantity acquired on the spot market. We will therefore introduce:

$$C_{BIAS} = \sum_{h=1}^{Nh} \left\{ P_{BM}(h) \left[\sum_{i=1}^{NC} E(i,h) - \sum_{s=1}^{Ns} B(s,h) \right] \right\} . \tag{4}$$

where $P_{BM}(h)$ is the hourly price in the balancing market. By definition, C_{BIAS} is a cost; clearly this holds when the provisions of the retail company in the spot market are lesser than the load effectively requested by final users. On the other hand, in case of demand drops, Eq.(4) will express a kind of negative cost.

(d) As a consequence of activities described in points (a) to (c), the retail company must perform some financial hedging to prevent (or better: to limit) portfolio risk:

$$H = \sum_{j=1}^{NFc} \sum_{h=1}^{Nh} \left[S(j,h) - P_{BM}(h) \right] \; Bh(j,h) . \tag{5}$$

where $S(j,h)$ is the strike price of the $j-$th financial contract used to hedge the retailer's position with respect to the $h-$th hourly profile, and $Bh(j,h)$ is the corresponding bandwidth (in MWh).

From Eqs. (2) to (5) it turns then out that the profit of a generic power retailer may be expressed as:

$$\pi = f(R_{SELL}, C_{SP}, C_{BIAS}, H) = R_{SELL} - C_{SP} - C_{BIAS} + H . \tag{6}$$

3 The Optimization Model

Holding the assumptions made in the previous section, the activity on the Balancing Market becomes a key issue for the retailer, as illustrated in Figure 1.

The Balancing Market is the place where the load settled through contracts on the spot market (solid black line on top graph) meets energy volumes demanded by consumers (dashed gray line on top graph): the difference among those quantities is what is effectively traded on such market (centered graph at the bottom of Figure 1), and it may led the retailer to attain either profits (shaded gray area above x–axes), or losses (shaded gray area below x–axes).

The maximization of retailer's profit is therefore a problem depending on the bandwiths of energy settled on the physical market, and hedged through financial

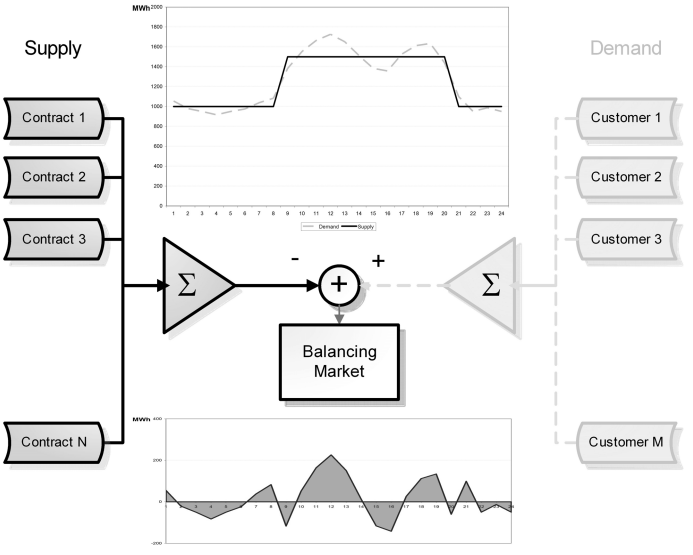


Fig. 1. Activity on the Balancing Market

contracts; the general optimization model for the retail company may be then written as follows:

$$\begin{aligned} &\max \pi \\ &s.t. \\ &0 \leq B(s,h) \leq B_{MAX}(s,h), s = 1,\ldots,Ns;\ h = 1,\ldots,Nh; &(7)\\ &0 \leq Bh(j,h) \leq Bh_{MAX}(j,h),\ j = 1,\ldots,NFc;\ h = 1,\ldots,Nh; &(8)\\ &CVaR_{\alpha}(\Pi) \leq RL. &(9) \end{aligned}$$

Eqs (7) to (9) define a multi–period optimization problem, faced by the retailer at each time h $(h = 1,\ldots,Nh)$, i.e. for each hourly profile.

In deepest detail, at each time h $(h = 1,\ldots,Nh)$, we will have Ns physical constraints of type (7). Those contraints rule out the energy bandwith settled with the generic s–th generation company to maintain non–negative, and to stay under a maximum threshold level $B_{MAX}(s,h)$. The NFc constraints of type (8), on the other hand, are the financial counterparts of (7): their presence makes sense only within a deregulated envinroment, where the retailer needs to hedge from perspective losses. Finally, constraints of type (9) are robustness constraints.

The novelty of our approach, at this point, stands primarily in the way robustness constraints are posed, since:

(i) they force the overall Conditional Value at Risk, at the confidence interval $1-\alpha$, to lay down the value RL that represents the maximum loss the retailer can sustain;

(ii) they assume that $CVaR_\alpha$ depends on the stochastic variable Π, being:

$$CVaR_\alpha(\Pi) = CV_\alpha \{\Pi[P_{BM}(h), E(i,h), B(s,h), Bh(j,h)]\}, \quad (10)$$
$$h = 1, \ldots, Nh;\ i = 1, \ldots, Nc;$$
$$s = 1, \ldots, Ns;\ j = 1, \ldots, NFc.$$

where $P_{BM}(h)$, $E(i,h)$, $B(s,h)$, and $Bh(j,h)$ are the variables already introduced and described in the previous section.

3.1 Expliciting CVaR Constraints

Type (9) constraints provide the general formulation of how $CVaR_\alpha$ enters into the optimization scheme. We now define:

$$\mathbf{PBM} = \{P_{BM}(h)\}_{h=1,\ldots,Nh}\ ; \quad (11)$$
$$\mathbf{En} = \{E(i,h)\}_{i=1,\ldots,Nc;\ h=1,\ldots,Nh}\ . \quad (12)$$

i.e. **PBM** is an array whose components are Nh realizations of the random variable *electricity price on the balancing market*, observed from $h = 1$ to Nh, and **En** is the $Nc \times Nh$ matrix whose elements are Nh realizations of the random variable *energy volume*, which represents the energy demanded by each group of final users. The rationale is that since prices vary along a frame horizon of length Nh, we must consider Nh different possible realizations for such variable. In a similar way, the retailer faces to Nc different types of consumers, and hence to Nc energy volumes that, like the price, vary along time too. Using Eqs. (11) and (12), we can then build L scenarios:

$$S_\ell = \{\mathbf{PBM}_\ell, \mathbf{En}_\ell\},\ \ell = 1, \ldots, L\ ;$$

with associated probability P_ℓ, being $\sum_{\ell=1}^{L} P_\ell = 1$. For sake of simplicity, here we assume that S_ℓ are uniformly distributed random variables, but, obviously, more sophisticated assumptions may be made. Hence, from (9) and (10) we derive the following $L \times Nh \times Nc \times NFc + 1$ constraints for $CVaR_\alpha$:

$$\nu + \sum_{\ell=1}^{L} P_\ell t_\ell \le RL\ ; \quad (13)$$

$$\frac{1}{\alpha}\{\Pi_\ell[B(s,h), Bh(j,h)] - \nu\} \le t_\ell,\ \forall \ell, s, j, h\ . \quad (14)$$

with $\nu \ge 0$, $\ell = 1, \ldots, L$, $s = 1, \ldots, Ns$; $j = 1, \ldots, NFc$, and $h = 1, \ldots, Nh$. In practice, this corresponds to explode Eq. (9) into two classes of constraints:

– the constraint given in Eq. (13), that fixes the contribution of each scenario, weighted by the probability it has to really occur;

– the L constraints given in Eq.(14), where the contribution of each contract is defined as the best scenario outcome, provided the energy bandwiths to be negotiated both in physical and financial market.

The robustness of constraints emerges in many ways. First of all, the suggested scheme tends by construction to mitigate the typical effect of optimization models to find solutions that are "extreme" allocations. Additionally, the formulation reveals particularly suitable to be adapted to multi–stage problems. This is a highly desirable feature within the framework under examination, where retailers manage physical and financial contracts at very different duration. Not less importantly, the suggested scheme may be built directly from data, and hence it is suitable to be data driven.

4 An Application

We now consider a toy simulation to give an idea about how the robust optimization scheme works. To such purpose, we are going to write anew the optimization problem for the retail company, giving an aggregated form to deterministic constraints provided by (7) and (8), and considering the *exploded* form of the robust constraint given in (9):

$$
\begin{aligned}
&\max \pi \\
&s.t. \\
&0 \leq \sum_{s=1}^{Ns} \sum_{h=1}^{Nh} B(s,h) && \leq \overline{B}; \\
&0 \leq \sum_{j=1}^{NFc} \sum_{h=1}^{Nh} Bh(j,h) && \leq \overline{Bh}; \\
&\nu + \sum_{\ell=1}^{L} P_\ell t_\ell && \leq RL; \\
&\tfrac{1}{\alpha}\{\Pi_\ell[B(s,h), Bh(j,h)] - \nu\} \leq t_\ell, \ \forall \ell, s, j, h \ .
\end{aligned}
$$

where $\overline{B}$ and $\overline{Bh}$ are the maximum energy volumes (in MWh) that the retailer can trade through physical and financial contracts, respectively.

Table 1 reports main features of the examined simulation.

In particular, the simulation involves robust constraints at 95% confidence interval level, a time horizon of one year ($Nh = 24 \times 365$), while the maximum exposure to loss has been set at 78000 Euro. It is also worthwhile to spend a few words on parameter L, that expresses the number of scenarios to build. The

Table 1. Simulation main features

α	Ns	Nh	NFc	Nc	L	$\overline{B}$ (MWh)	$\overline{Bh}$ (MWh)	RL Euro
5%	10	8760	20	10	7	5000	3000	78000

value of L is apparently too low, but it is fully consistent with the perspectives and the needs of typical retailers in real world allocation problems. As a matter of fact, this is the very key issue, that justifies the adoption of a robust tractation to the optimization problem: whereas the probabilistic approach needs a huge amount of data to build proper statistical description of problem parameters, our approach is data–driven, and it works with limited available knowledge, allowing to create scenarios that are nothing but perturbations around the available historical values of the observed variables. This made us possible to generate $S_\ell = \{\mathbf{PBM}_\ell, \mathbf{En}_\ell\}, (\ell = 1, \ldots, L)$, using historical data, and perturbing them with a fraction $1/L$ of their overall variance.

Table 2 shows the results in the discussed case, and the comparison to those obtained running a deterministic scheme without the insertion of any robust constraint.

Table 2. Simulation results

	Det. Sol.	Rob. Sol.	Delta %
RL (Cvar)= 78000 Euro			
Average Profit [Euro]	8 603 656.11	8 576 042.59	-0.32%
CVaR [Euro]	101 473.00	78 000.00	-23.33%

Viewing at Table 2, one can immediately observe that the robust scheme allows to bound the maximum loss to which the retailer is exposed: while RL is inflated as a constraint into the robust problem, in the deterministic case the exposure to risk is evaluated *a posteriori*, once the optimal solution has been found, and it is generally higher than in the robust case. In the examined situation, for instance, the robust solution leads to a reduction of 0.32% in the average profit, but the deterministic solution requires from the retailer an exposure sensitively greater than in the robust case (+23.33%).

5 Conclusive Remarks and Future Directions

We presented an optimization scheme which allows a realistic representation of the allocation problem faced by retail companies which operate on deregulated power markets.

Prior to restructuring, consumers were charged an all–inclusive price, covering all aspects of utility service (generation, transmission, delivery, metering, billing, and any ancillary costs). As energy markets have evolved in the late 1990s away from cost–based transactions to competitive market–based transactions, the exposure to market risks for the variable cost of supply has substantially increased. As main consequence, the retailers in order to maximize their profit must determine the optimal mix of energy bandwidth they have to trade both through physical contract, and through financial contracts, to hedge from major risk exposure.

Starting from this point, we have suggested a robust optimization framework that makes retailers capable to manage both physical and financial risk, by

creating joint scenarios of price–energy profiles. Actually, to the best of our knowledge, this approach seems rather unexplored, and it is surely original in the way robust constraints are explicited by way of Conditional Value at Risk. The main advantages of such approach rely inside its tractability, as well as in the possibility to be easily customizable to replicate significant observable conditions.

More generally, with the scenarios approach inflated into the robust optimization scheme, it is possible to assure a better awareness of the examined problem in all its (uncertainty) aspects. Not less importantly, the suggested scheme may be built directly from data, and hence it is suitable to be data driven.

Presently some unaddressed questions remain, although we have planned to make them objects of future works:

(i) the convenience of the robust approach with respect to more traditional counterparts, like stochastic and dynamic programming. Such convenience, in particular, need to be explored under two different aspects: the reduction of total performance, and the improvement in downside risk protection;
(ii) the theoretical characterization of the price of robustness in term of optimality.

References

1. Ben-Tal, A., Nemirovski, A.: Robust Solutions of Linear Programming Problems Contaminated with Uncertain Data. Math. Progr. J. 88(3), 411–424 (2000)
2. Ben-Tal, A., Margalit, T., Nemirovski, A.: Robust Modeling of Multi-Stage Portfolio Problems. In: Frenk, H., Roos, K., Terlaky, T., Zhang, S. (eds.) High Performance Optimizatio. Applied Optimization series, vol. 33, pp. 303–328. Kluwer Academic, Dordrecht (2000)
3. Ben-Tal, A., Nemirovski, A.: Robust Optimization - Methodology and Applications. Math. Progr. J. 92(3), 453–480 (2002)
4. Ben-Tal, A., Nemirovski, A.: Selected Topics in Robust Convex Optimization. Math. Progr. J. 112(1), 125–158 (2008)
5. Bertsimas, D., Brown, D., Caramanis, C.: Theory and Applications of Robust Optimization. Technical Report (July 2007)
6. Ben-Tal, A., Bertsimas, D., Brown, D.: A Flexible Approach to Robust Optimization via Convex Risk Measures. Technical Report (September 2006)
7. Dorris, G., Burrows, S.: Retail Risk–Based Pricing, A new approach to rate design. Public Utilities Fortnightly 142(3) (2004)
8. El–Khattam, W., Bhattacharya, K., Hegazy, Y., Salama, M.M.A.: Optimal Investment Planning for Distributed Generation in a Competitive Electricity Market. IEEE Trans. on Power Systems 19(3), 1674–1684 (2004)
9. Gomez Villalva, E., Ramos, A.: Optimal Energy Management of an Industrial Consumer in Liberalized Markets. IEEE Trans. on Power Systems 18(2), 716–726 (2003)
10. Lewiner, C.: Business and Technology Trends in the Global Utility Industries. IEEE Power Eng. Rev. 21(12), 7–9 (2001)
11. Markowitz, H.M.: Portfolio Selection. J. of Finance 7(1), 77–91 (1952)
12. Soyster, A.L.: Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming. Op. Res. 21(5), 1154–1157 (1973)

Value Functions and Transversality Conditions for Infinite Horizon Optimal Control Problems

Nobusumi Sagara*

Faculty of Economics, Hosei University
4342, Aihara, Machida, Tokyo, 194–0298, Japan
nsagara@hosei.ac.jp

Abstract. This paper investigates infinite horizon optimal control problems with fixed left endpoints with nonconvex, nonsmooth data. We derive the nonsmooth maximum principle and the adjoint inclusion for the value function as necessary conditions for optimality that indicate the relationship between the maximum principle and dynamic programming. The necessary conditions under consideration are extensions of those of [8] to an infinite horizon setting. We then present new sufficiency conditions consistent with the necessary conditions, which are motivated by the useful result by [26] whose sufficiency theorem is valid for nonconvex, nondifferentiable Hamiltonians. The sufficiency theorem presented in this paper employs the strengthened adjoint inclusion of the value function as well as the strengthened maximum principle. If we restrict our result to convex models, it is possible to characterize minimizing processes and provide necessary and sufficient conditions for optimality. In particular, the role of the transversality conditions at infinity is clarified.

Keywords: Nonsmooth maximum principle; Infinite horizon; Value function; Sufficiency; Transversality condition.

1 Necessary Condition for Optimality

Let $[0,\infty)$ be a half-open interval of the real line with the σ-algebra $\mathscr{L}$ of Lebesgue measurable subsets of $[0,\infty)$. Denote the product σ-algebra of $\mathscr{L}$ and the product σ-algebra $\mathscr{B}^m \times \mathscr{B}^n$ of Borel subsets of $\mathbb{R}^m \times \mathbb{R}^n$ by $\mathscr{L} \times \mathscr{B}^m \times \mathscr{B}^n$. We are given $\mathscr{L} \times \mathscr{B}^m \times \mathscr{B}^n$-measurable functions $L : [0,\infty) \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ and $f : [0,\infty) \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$, an $\mathscr{L} \times \mathscr{B}^m$-measurable subset Ω of $[0,\infty) \times \mathbb{R}^m$ and a set-valued mapping $U : [0,\infty) \rightrightarrows \mathbb{R}^n$. The t-section of Ω is denoted by $\Omega(t)$, that is, $\Omega(t) = \{x \in \mathbb{R}^m \mid (t,x) \in \Omega\}$ for $t \in [0,\infty)$. The optimal control

* This research was partly supported by a Grant-in-Aid for Scientific Research (No. 18610003) from the Japan Society for the Promotion of Science.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 273–282, 2008.

problem under investigation is the following one:

$$\begin{aligned} \min\ & J(x,u) := \int_0^\infty L(t,x(t),u(t))dt \\ \text{s.t. } & \dot{x}(t) = f(t,x(t),u(t)) \quad \text{a.e. } t \in [0,\infty), \\ & x(0) = x, \\ & u(t) \in U(t) \quad \text{a.e. } t \in [0,\infty), \\ & x(t) \in \Omega(t) \quad \text{for every } t \in [0,\infty). \end{aligned} \tag{P_0^∞}$$

Here the minimization is taken over all locally absolutely continuous functions $x : [0,\infty) \to \mathbb{R}^m$ and $\mathscr{L}$-measurable functions $u : [0,\infty) \to \mathbb{R}^n$ satisfying the constraint for the problem (P_0^∞).

An ε-*tube* about the continuous function $x : [0,\infty) \to \mathbb{R}^m$ is a set of the form

$$T^\varepsilon = \{(t,x) \in \Omega \mid x \in x(t) + \varepsilon B\}$$

with $\varepsilon > 0$, where B is the open unit ball in $\mathbb{R}^m$. A *process* on a given subinterval I of $[0,\infty)$ is a pair of functions $(x(\cdot),u(\cdot))$ on I of which $x : I \to \mathbb{R}^m$ is locally absolutely continuous functions and $u : I \to \mathbb{R}^n$ is a measurable function such that the constraints (when I replaces $[0,\infty)$) except for the initial condition in problem (P_0^∞) are satisfied and the integrand $L(\,\cdot\,,x(\cdot),u(\cdot))$ is integrable on I. A process $(x(\cdot),u(\cdot))$ on I is *admissible* if $x(t_0) = x$, where t_0 is the left endpoint of I. A process on I is *minimizing* if it minimizes the value of the integral functional $\int_I L dt$ over all admissible processes on I. When $I = [0,\infty)$, we shall abbreviate the domain on which processes are defined. In this section $(x_0(\cdot),u_0(\cdot))$ is taken to be a fixed minimizing process on $[0,\infty)$ for (P_0^∞) for which $x_0(\cdot)$ is contained in some tube.

Now define the value function $V : \Omega \to \mathbb{R} \cup \{\pm\infty\}$ by

$$V(t,x) = \inf \int_t^\infty L(s,x(s),u(s))ds,$$

where the infimum is taken over all processes $(x(\cdot),u(\cdot))$ on $[t,\infty)$ for which $x(t) = x \in \Omega(t)$. When no such processes exist, the value is supposed to be $+\infty$ as usual. Throughout this paper, the generalized gradient of $V(t,\cdot)$ at x is denoted by $\partial V(t,x)$.

1.1 Maximum Principle with an Infinite Horizon

For notational simplicity, define the function $\tilde{f}$ by

$$\tilde{f} = \begin{pmatrix} L \\ f \end{pmatrix} : [0,\infty) \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^{m+1}.$$

The basic hypotheses to derive necessary conditions for optimality are as follows:

Hypothesis 1. (i) graph (U) is $\mathscr{L} \times \mathscr{B}^n$-measurable;
(ii) $\tilde{f}(t,\cdot,u)$ is Lipschitz of rank $k(t)$ on $\Omega(t)$ for every $(t,u) \in$ graph (U) with $k(\cdot)$ integrable on $[0,\infty)$ and $\tilde{f}(\,\cdot\,,x,\cdot\,)$ is $\mathscr{L} \times \mathscr{B}^n$-measurable for every $x \in \mathbb{R}^m$;

(iii) There exists an integrable function φ on $[0,\infty)$ such that

$$|L(t,x_0(t),u)| \le \varphi(t) \quad \text{for every } (t,u) \in \operatorname{graph}(U);$$

(iv) There exists an ε-tube $\Omega' \subset \Omega$ about $x_0(\cdot)$ such that $V(t,\cdot\,,)$ is Lipschitz of rank K on $\Omega'(t)$ for every $t \in [0,\infty)$.

The Lipschitz continuity of the value function in the condition (iv) of the hypothesis is very mild if the existence of minimizing processes is guaranteed. In fact it is almost superfluous, since one is most often interested in examples of the optimal control problem in which $\Omega = [0,\infty) \times \mathbb{R}^m$; As seen in Section 3, the condition (iv) of the hypothesis is implied from the other conditions.

The Pontryagin (or pseudo) Hamiltonian for (P_0^∞) is given by

$$H_P(t,x,u,p) = \langle p, f(t,x,u)\rangle - L(t,x,u).$$

Theorem 2. *Suppose that Hypothesis 1 is satisfied. Then there exists a locally absolutely continuous function $p : [0,\infty) \to \mathbb{R}^m$ with the following properties:*

(i) $-\dot{p}(t) \in \partial_x H_P(t,x_0(t),u_0(t),p(t))$ *a.e.* $t \in [0,\infty)$;
(ii) $H_P(t,x_0(t),u_0(t),p(t)) = \max_{u\in U(t)} H_P(t,x_0(t),u,p(t))$ *a.e.* $t \in [0,\infty)$;
(iii) $-p(t) \in \partial V(t,x_0(t))$ *a.e.* $t \in [0,\infty)$;
(iv) $-p(0) \in \partial V(0,x)$.

It should be noted that the maximum principle is relevant to the (true) Hamiltonian for (P_0^∞) given by

$$H(t,x,p) = \sup_{u\in U(t)} \{\langle p, f(t,x,u)\rangle - L(t,x,u)\}.$$

Corollary 1. *The condition (i) of Theorem 2 implies that*

$$\dot{p}(t) \in \partial_x H(t,x_0(t),p(t)) \quad \text{a.e. } t \in [0,\infty).$$

Note that the assertions of the theorem do not exclude the possibility that $-p(t) \notin \partial V(t,x_0(t))$ for every t in some null set in $[0,\infty)$. We seek then additional hypotheses under which the condition (iii) of the theorem can be strengthened to

$$-p(t) \in \partial V(t,x_0(t)) \quad \text{for every } t \in [0,\infty). \tag{1}$$

To this end, the argument of [8] is valid to the infinite horizon setting for obtaining the following result.

Corollary 2. *The condition (iii) of Theorem 2 can be strengthened to the condition (1) if the set-valued mapping $\partial V(\,\cdot\,,x_0(\cdot)) : [0,\infty) \rightrightarrows \mathbb{R}^m$ is upper semicontinuous.*

2 Properties of the Value Function

We have assumed in Hypothesis 1(iv) that the value function is Lipschitz continuous. In Subsection 3.1, we demonstrate the Lipschitz continuity of the value function under the existence of minimizing process for any initial point in time and initial condition. For the finite horizon case, the result is well-known, but some intricate arguments are involved for the infinite horizon case because the finite horizon value functions are approximated to the infinite horizon value function.

The convexity of the value function is proven in Subsection 3.2 under some additional assumptions. It simplifies the adjoint inclusion for the value function in Theorem 2 to the usual subgradient inclusions and enables one to characterize minimizing process along with the concavity of the Hamiltonian as developed in Section 5.

2.1 Lipschitz Continuity of the Value Function

Hypothesis 3. For every $(t,x)\in\Omega$ there exists a minimizing process $(\hat{x}(\cdot),\hat{u}(\cdot))$ on $[t,\infty)$ for the problem

$$\begin{aligned}
&\min \int_t^\infty L(s,x(s),u(s))ds\\
&\text{s.t. } \dot{x}(s)=f(s,x(s),u(s)) \quad \text{a.e. } s\in[t,\infty),\\
&\qquad x(t)=x,\\
&\qquad u(s)\in U(s) \quad \text{a.e. } s\in[t,\infty),\\
&\qquad x(s)\in\Omega(s) \quad \text{for every } s\in[t,\infty)
\end{aligned} \tag{P_t^∞}$$

such that some ε-tube about $\hat{x}(\cdot)$ is contained in Ω.

To guarantee the existence of a minimizing process for (P_t^∞), one needs convexity assumptions on $\tilde{f}$ as in [2,3,4,5]. However, the existence of the minimizing process $(x_0(\cdot),u_0(\cdot))$ for (P_0^∞) implies that the above hypothesis is innocuous in the following typical case in which $\tilde{f}$ is autonomous (independent of s), $U(s)\equiv\hat{U}\subset\mathbb{R}^n$ and $\Omega(s)\equiv\hat{\Omega}\subset\mathbb{R}^m$. To see this, note that the process on $[t,\infty)$ defined by

$$(\hat{x}(s),\hat{u}(s))=(x_0(t-s),u_0(t-s)) \quad \text{for } s\in[t,\infty)$$

is a minimizing process for problem (P_t^∞).

Theorem 4. *Suppose that Hypotheses 1 and 3 is satisfied. Then V is measurable on Ω and $V(t,\cdot)$ is Lipschitz of rank K on $\Omega(t)$ for every $t\in[0,\infty)$.*

2.2 Convexity of the Value Function

Define the set-mapping $\tilde{\Gamma}:\Omega\to\mathbb{R}\times\mathbb{R}^m$ by

$$\tilde{\Gamma}(t,x)=\{(z,v)\in\mathbb{R}\times\mathbb{R}^m \mid \exists u\in U(t): z\geq L(t,x,u),\ v=f(t,x,u)\}.$$

Hypothesis 5. (i) $\Omega(t)$ is convex for every $t \in [0, \infty)$.
(ii) $\tilde{\Gamma}(t, \cdot) : \Omega(t) \rightrightarrows \mathbb{R}^m$ has the convex graph for every $t \in [0, \infty)$.

Hypothesis 5 is somewhat stronger than the standard convexity hypothesis guaranteeing the existence of a minimizing process that $\tilde{\Gamma}(t, \cdot)$ is convex-valued for every $t \in [0, \infty)$. (See [2,3,4,5].) One of the sufficient conditions guaranteeing Hypothesis 5 is the following:

Hypothesis 6. (i) $\Omega(t) \times U(t)$ are convex for every $t \in [0, \infty)$;
(ii) $L(t, \cdot, \cdot)$ is convex on $\Omega(t) \times U(t)$ for every $t \in [0, \infty)$.
(iii) $f(t, \cdot, U(t)) : \Omega(t) \rightrightarrows \mathbb{R}^m$ has the convex graph for every $t \in [0, \infty)$;

Theorem 7. *$V(t, \cdot)$ is convex on $\Omega(t)$ for every $t \in [0, \infty)$ if Hypotheses 1, 3 and 5 are satisfied.*

Corollary 3. *The condition (iii) of Theorem 2 is strengthened to the condition (11) if Hypotheses 1, 3 and 5 are satisfied.*

3 Sufficient Conditions for Optimality

We now turn for the important issue of *sufficient conditions*: conditions that assure that a given admissible process is in fact a solution of the problem. In this section we present two sufficiency theorems: The first one is an extension of the result by [26] to the infinite horizon setting that is consistent with the maximum principle. The second one is novel in the literature in that the sufficient condition is related to the adjoint inclusion of the value function and it is consistent with the necessary condition of Theorem 2.

3.1 Sufficiency Theorems

Definition 1. *An admissible process $(x_0(\cdot), u_0(\cdot))$ for (P_0^∞) is a locally minimizing process if there exists some $\varepsilon > 0$ such that $(x_0(\cdot), u_0(\cdot))$ minimizes the functional $J(x, u)$ over all admissible processes $(x(\cdot), u(\cdot))$ satisfying $x(t) \in x_0(t) + \varepsilon B$ for every $t \in [0, \infty)$.*

Hypothesis 8. (i) $L(t, \cdot, \cdot)$ is lower semicontinuous on $\Omega(t) \times U(t)$ for every $t \in [0, \infty)$.
(ii) $f(t, \cdot, \cdot)$ is continuous on $\Omega(t) \times U(t)$ for every $t \in [0, \infty)$.
(iii) $U(t)$ is closed for every $t \in [0, \infty)$ and graph (U) is $\mathscr{L} \times \mathscr{B}^n$-measurable.
(iv) For every $t \in [0, \infty)$ and for every bounded subset S of $\mathbb{R}^m \times \mathbb{R}^m$, the set

$$\{u \in U(t) \mid \exists (x, v) \in S : f(t, x, u) = v\}$$

is bounded.

Theorem 9. *Suppose that Hypothesis 8 is satisfied. Let $(x_0(\cdot), u_0(\cdot))$ be an admissible process for (P_0^∞) such that the ε-tube about $x_0(\cdot)$ is contained in Ω. Suppose that there exist a locally absolutely continuous function $p : [0, \infty) \to \mathbb{R}^m$ and a locally absolutely continuous $m \times m$-symmetric matrix-valued function P on $[0, \infty)$ with the following properties:*

(i) *For every* $v \in \varepsilon B$ *and* $u \in U(t)$*:*

$$\begin{aligned} &H_P(t, x_0(t) + v, u, p(t) - P(t)v) \\ &\leq H_P(t, x_0(t), u_0(t), p(t)) - \langle \dot{p}(t) + P(t)\dot{x}_0(t), v\rangle + \tfrac{1}{2}\langle v, \dot{P}(t)v\rangle \end{aligned}$$

a.e. $t \in [0, \infty)$*;*

(ii) *For every* $\eta > 0$ *there exists some* $t_0 \in [0, \infty)$ *such that* $t \geq t_0$ *implies*

$$\tfrac{1}{2}\langle v, P(t)v\rangle < \langle p(t), v\rangle + \eta \quad \textit{for every } v \in \varepsilon B.$$

Then $(x_0(\cdot), u_0(\cdot))$ *is a minimizing process for* (P_0^∞).

Note that the condition (i) of the theorem implies the conditions (i) and (ii) of Theorem 2. When the matrix-valued function P in the theorem happens to be identically the zero matrix, the condition (i) of the theorem reduces to the subgradient inequality for H:

$$H(t, x_0(t) + v, p(t)) - H(t, x_0(t), p(t)) \leq -\langle \dot{p}(t), v\rangle \tag{2}$$

for every $v \in \mathbb{R}^m$. The condition (2) is imposed also by [9] to obtain the sufficiency result. This is, of course, satisfied if $H(t, x, p(t))$ is concave in x for every $t \in [0, \infty)$. Thus, the condition (i) of the theorem can be viewed as a strengthening of the necessary condition (i) of Theorem 2. Moreover, if $P = 0$, then the condition (ii) of the theorem is equivalent to the transversality condition at infinity:

$$\lim_{t\to\infty} p(t) = 0.$$

For the differentiable case, sufficient conditions for optimality were given by [11] under the hypothesis that the Hamiltonian H_P is concave in (x, u), whose result was extended by [20]. Thus, the above observation leads to an extension of the Mangasarian sufficiency theorem with an infinite horizon:

Corollary 4. *Suppose that Hypothesis 8 is satisfied and* $\Omega(t)$ *is convex for every* $t \in [0, \infty)$*. Let* $(x_0(\cdot), u_0(\cdot))$ *be an admissible process for* (P_0^∞) *such that the* ε*-tube about* $x_0(\cdot)$ *is contained in* Ω *and* $p : [0, \infty) \to \mathbb{R}^m$ *be a locally absolutely continuous function with the following properties:*

(i) $H(t, \cdot, p(t))$ *is concave on* $\Omega(t)$ *for every* $t \in [0, \infty)$*;*
(ii) $-\dot{p}(t) \in \partial_x H(t, x_0(t), p(t))$ *a.e.* $t \in [0, \infty)$*;*
(iii) $H_P(t, x_0(t), u_0(t), p(t)) = H(t, x_0(t), p(t))$ *a.e.* $t \in [0, \infty)$*;*
(iv) $\lim_{t\to\infty} p(t) = 0$.

Then $(x_0(\cdot), u_0(\cdot))$ *is a minimizing process for* (P_0^∞).

Consider the following transversality condition at infinity:

$$\limsup_{t\to\infty} \langle p(t), x(t) - x_0(t)\rangle \geq 0 \tag{3}$$

for every admissible arc for (P_0^∞). To obtain the sufficiency result, [20] imposed the condition (3) in addition to the conditions (i) and (ii) of the corollary as well as the differentiability assumption on $\tilde{f}$ and [9] assumed (3) for the nondifferentiable nonconcave Hamiltonians along with the condition (2).

Note that the condition (2) is implied by the condition (iv) of the corollary if every admissible arc is bounded. However, if admissible arcs are unbounded, this condition is difficult to check in practice because it involves possible information on the limit behavior of all admissible arcs. The condition (iv) of the corollary on its own right needs no such information. For the derivation of it as a necessary condition, see [1,12].

We provide a new sufficient condition in terms of the value function. Contrary to Theorem 9, it does not need Hypothesis 8 and is consistent with the necessary condition in Theorem 2.

Theorem 10. *Let $(x_0(\cdot), u_0(\cdot))$ be an admissible process for (P_0^∞) such that the ε-tube about $x_0(\cdot)$ is contained in Ω. Suppose that there exist a locally absolutely continuous function $p : [0,\infty) \to \mathbb{R}^m$ and a locally absolutely continuous $m \times m$-symmetric matrix-valued function P on $[0,\infty)$ with the following properties:*

(i) *For every $v \in \varepsilon B$ and $u \in U(t)$:*

$$\begin{aligned} &H_P(t, x_0(t) + v, u, p(t) - P(t)v) \\ &\le H_P(t, x_0(t), u_0(t), p(t)) - \langle \dot{p}(t) + P(t)\dot{x}_0(t), v\rangle + \tfrac{1}{2}\langle v, \dot{P}(t)v\rangle \end{aligned}$$

a.e. $t \in [0,\infty)$;

(ii) *For every $v \in \varepsilon B$:*

$$V(t, x_0(t) + v) \le V(t, x_0(t)) - \langle p(t) + P(t)x(t), v\rangle + \tfrac{1}{2}\langle v, P(t)v\rangle;$$

(iii) $\lim_{t\to\infty} V(t, x_0(t)) = 0.$

Then $(x_0(\cdot), u_0(\cdot))$ is a minimizing process for (P_0^∞).

Note that when $P = 0$ in the theorem, the condition (ii) of the theorem reduces to the subgradient inequality for $V(t, \cdot)$:

$$V(t, x_0(t) + v) - V(t, x_0(t)) \le -\langle p(t), v\rangle$$

for every $v \in \mathbb{R}^m$. This is, indeed, satisfied if $V(t, x)$ is convex in x for every $t \in [0,\infty)$. Thus, the conditions (ii) of the theorem can be viewed as a strengthening of the necessary condition (ii) of Theorem 2.

Corollary 5. *Suppose that $\Omega(t)$ is convex for every $t \in [0,\infty)$ and $V(t, \cdot)$ is convex on $\Omega(t)$ for every $t \in [0,\infty)$. Let $(x_0(\cdot), u_0(\cdot))$ be an admissible process for (P_0^∞) such that the ε-tube about $x_0(\cdot)$ is contained in Ω and $p : [0,\infty) \to \mathbb{R}^m$ be a locally absolutely continuous function with the following properties:*

(i) $H(t, \cdot, p(t))$ *is concave on* $\Omega(t)$ *for every* $t \in [0, \infty)$;
(ii) $-\dot{p}(t) \in \partial_x H(t, x_0(t), p(t))$ *a.e.* $t \in [0, \infty)$;
(iii) $H_P(t, x_0(t), u_0(t), p(t)) = H(t, x_0(t), p(t))$ *a.e.* $t \in [0, \infty)$;
(iv) $-p(t) \in \partial_x V(t, x_0(t))$ *for every* $t \in [0, \infty)$;
(v) $\lim_{t\to\infty} V(t, x_0(t)) = 0$.

Then $(x_0(\cdot), u_0(\cdot))$ *is a minimizing process for* (P_0^∞).

The role of the limit behavior of the value function in the condition (v) of the corollary was also exploited by [6] and [22] in the derivation of the sufficiency result for the convex problem of calculus of variations.

4 Necessary and Sufficient Condition for Optimality

In this section we derive the necessary and sufficient conditions for optimality under convexity hypotheses. For the finite horizon case, [16] obtained necessary and sufficient conditions for optimality under convexity hypotheses. The convex model examined here clarifies the role of the Hamiltonian and the value function for a complete characterization of optimality. Furthermore, we investigate the role of transversality conditions at infinity and derive them as a necessary and sufficient condition for optimality.

4.1 Concavity of the Hamiltonian

The concavity of the Hamiltonian is subtler than the convexity of the value function. Strictly speaking, Hypothesis 6 guaranteeing the convexity of the value function $V(t, x)$ in x is insufficient to establish the concavity of the Hamiltonian $H(t, x, p)$ in x.

Note that for every $(t, x) \in \Omega$ and $p \in \mathbb{R}^m$:

$$H(t, x, p) = \sup_{v\in\mathbb{R}^m} \{\langle p, v\rangle - F(t, x, v)\}.$$

Thus, $H(t, x, p)$ is concave in x if $F(t, x, v)$ is convex in (x, v). However, the convexity of $F(t, \cdot, \cdot)$ does not necessarily follows from Hypothesis 6. To overcome this difficulty, we must strengthen Hypothesis 6 according to [9].

Hypothesis 11. (i) $\Omega(t) \times U(t)$ is convex for every $t \in [0, \infty)$;
(ii) $L(t, \cdot, \cdot)$ is convex on $\Omega(t) \times U(t)$ for every $t \in [0, \infty)$ and $L(t, x, \cdot)$ is nondecreasing on $U(t)$ for every $(t, x) \in \Omega$;
(iii) $f(t, \cdot, \cdot) : \Omega(t) \times U(t) \to \mathbb{R}^n$ is concave for every $t \in [0, \infty)$;
(iv) $f(t, \cdot, U(t)) : \Omega(t) \rightrightarrows \mathbb{R}^m$ has the convex graph for every $t \in [0, \infty)$;
(v) For every $v \in f(t, x, U(t))$ and $u \in U(t)$ with $v \le f(t, x, u)$ and $x \in \Omega(t)$ there exists some $u' \in U(t)$ such that $u' \le u$ and $v = f(t, x, u')$.

As shown by [9], Hypothesis 11 is sufficient for $F(t, \cdot, \cdot)$ to be a convex function on $\Omega(t) \times \mathbb{R}^m$ for every $t \in [0, \infty)$, from which the concavity of the Hamiltonian follows.

Theorem 12. *$H(t,\cdot\,,p)$ is concave on $\Omega(t)$ for every $(t,p)\in[0,\infty)\times\mathbb{R}^m$ if Hypothesis 11 is satisfied.*

Corollary 6. *Suppose that Hypothesis 11 is satisfied. An admissible process $(x_0(\cdot),u_0(\cdot))$ is a minimizing process for (P_0^∞) if and only if the following conditions are satisfied:*

(i) *There exists a locally absolutely continuous function $p:[0,\infty)\to\mathbb{R}^m$ such that:*
  (a) $-\dot p(t)\in\partial_x H(t,x_0(t),p(t))$ *a.e.* $t\in[0,\infty)$;
  (b) $H_P(t,x_0(t),u_0(t),p(t))=H(t,x_0(t),p(t))$ *a.e.* $t\in[0,\infty)$;
  (c) $-p(t)\in\partial V(t,x_0(t))$ *for every* $t\in[0,\infty)$;
(ii) $\lim\limits_{t\to\infty}V(t,x_0(t))=0$.

4.2 Transversality Condition at Infinity

Hypothesis 13. (i) $\Omega(t)\subset\mathbb{R}^m_+$ for every $t\in[0,\infty)$;
(ii) $0\in U(t)$ a.e. $t\in[0,\infty)$;
(iii) $f(t,0,0)=0$ a.e. $t\in[0,\infty)$;
(iv) $L(t,0,0)\le 0$ a.e. $t\in[0,\infty)$;
(v) $L(t,\cdot\,,u)$ is nondecreasing on $\Omega(t)$ for every $u\in U(t)$ a.e. $t\in[0,\infty)$.

Corollary 7. *Suppose that Hypotheses 11 and 13 are satisfied. An admissible process $(x_0(\cdot),u_0(\cdot))$ is a minimizing process for (P_0^∞) if and only if there exists a locally absolutely continuous function $p:[0,\infty)\to\mathbb{R}^m$ such that:*

(i) $-\dot p(t)\in\partial_x H(t,x_0(t),p(t))$ *a.e.* $t\in[0,\infty)$;
(ii) $H_P(t,x_0(t),u_0(t),p(t))=H(t,x_0(t),p(t))$ *a.e.* $t\in[0,\infty)$;
(iii) $-p(t)\in\partial V(t,x_0(t))$ *for every* $t\in[0,\infty)$;
(iv) $\lim\limits_{t\to\infty}\langle p(t),x_0(t)\rangle=0$.

References

1. Aseev, S.M., Kryaziimskiy, A.: The Pontryagin Maximum Principle and Transversality Conditions for a Class of Optimal Control Problems with Infinite Time Horizons. SIAM J. Control Optim. 43, 1094–1119 (2004)
2. Balder, E.J.: An Existence Result for Optimal Economic Growth Problems. J. Math. Anal. Appl. 95, 195–213 (1983)
3. Bates, G.R.: Lower Closure Existence Theorems for Optimal Control Problems with Infinite Horizon. J. Optim. Theory Appl. 24, 639–649 (1978)
4. Baum, R.F.: Existence Theorems for Lagrange Control Problems with Unbounded Time Domain. J. Optim. Theory Appl. 19, 89–116 (1976)
5. Bell, M.L., Sargent, R.W.H., Vinter, R.B.: Existence of Optimal Controls for Continuous Time Infinite Horizon Problems. Internat. J. Control 68, 887–896 (1997)
6. Benveniste, L.M., Scheinkman, J.A.: Duality Theory for Dynamic Optimization Models of Economics: The Continuous Time Case. J. Econ. Theory 27, 1–19 (1982)
7. Clarke, F.H.: Optimization and Nonsmooth Analysis. John Wiley & Sons, New York (1983)

8. Clarke, F.H., Vinter, R.B.: The Relationship between the Maximum Principle and Dynamic Programming. SIAM J. Control Optim. 25, 1291–1311 (1987)
9. Feinstein, C.D., Luenberger, D.G.: Analysis of the Asymptotic Behavior of Optimal Control Trajectories: The Implicit Programming Problem. SIAM J. Control Optim. 19, 561–585 (1981)
10. Halkin, H.: Necessary Conditions for Optimal Control Problems with Infinite Horizon. Econometrica 42, 267–272 (1974)
11. Mangasarian, O.L.: Sufficient Conditions for the Optimal Control of Nonlinear Systems. SIAM J. Control Optim. 4, 139–151 (1966)
12. Michel, P.: On the Transversality Condition in Infinite Horizon Optimal Problems. Econometrica 50, 975–984 (1982)
13. Mlynarska, E.: Dual Sufficient Optimality Conditions for the Generalized Problem of Bolza. J. Optim. Theory Appl. 104, 427–442 (2000)
14. Nowakowski, A.: The Dual Dynamic Programming. Proc. Amer. Math. Soc. 116, 1089–1096 (1992)
15. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mischenko, E.F.: The Mathematical Theory of Optimal Processes. John Wiley & Sons, New York (1962)
16. Rockafellar, R.T.: Conjugate Convex Functions in Optimal Control and the Calculus of Variations. J. Math. Anal. Appl. 32, 174–222 (1970)
17. Rockafellar, R.T.: Optimal Arcs and the Minimum Value Function in Problems of Lagrange. Trans. Amer. Math. Soc. 180, 53–83 (1973)
18. Rockafellar, R.T.: Existence Theorems for Generalized Control Problems of Bolza and Lagrange. Adv. Math. 15, 312–333 (1975)
19. Sagara, N.: Nonconvex Variational Problem with Recursive Integral Functionals in Sobolev Spaces: Existence and Representation. J. Math. Anal. Appl. 327, 203–219 (2007)
20. Seierstadt, A., Sydsæter, K.: Sufficient Conditions in Optimal Control Theory. Internat. Econ. Rev. 18, 367–391 (1977)
21. Takekuma, S.-I.: Support Price Theorem for the Continuous Time Model of Capital Accumulation. Econometrica 50, 427–442 (1982)
22. Takekuma, S.-I.: On Duality Theory for the Continuous Time Model of Capital Accumulation. Hitotsubashi J. Econ. 25, 145–154 (1984)
23. Vinter, R.B.: New Results on the Relationship between Dynamic Programming and the Maximum Principle. Math. Control, Signals Systems 1, 97–105 (1988)
24. Ye, J.J.: Nonsmooth Maximum Principle for Infinite–Horizon Problems. J. Optim. Theory Appl. 76, 485–500 (1993)
25. Zeidan, V.: A modified Hamilton–Jacobi Approach in the Generalized Problem of Bolza. Appl. Math. Optim. 11, 97–109 (1984)
26. Zeidan, V.: First and Second Order Sufficient Conditions for Optimal Control and the Calculus of Variations. Appl. Math. Optim. 11, 209–226 (1984)
27. Zeidan, V.: New Second-Order Optimality Conditions for Variational Problems with C^2-Hamiltonians. SIAM J. Control Optim. 40, 577–609 (2000)

Modeling the Mobile Oil Recovery Problem as a Multiobjective Vehicle Routing Problem

Andréa C. Santos[1], Christophe Duhamel[1], and Dario J. Aloise[2]

[1] LIMOS, Université Blaise Pascal, complexe scientifique des Cézeaux, 63173, Aubière cedex, France
{andrea,christophe.duhamel}@isima.fr
[2] Federal University of Rio Grande do Norte, Campus Universitrio S/N, Lagoa Nova, 59072-970, Caixa-Postal 1551, Natal, RN, Brasil
dario@digizap.com.br

Abstract. The Mobile Oil Recovery (MOR) unit is a truck able to pump marginal wells in a petrol field. The goal of the MOR optimization Problem (MORP) is to optimize both the oil extraction and the travel costs. We describe several formulations for the MORP using a single vehicle or a fleet of vehicles. We have also strengthened them by improving the subtour elimination constraints. Optimality is proved for instances close to reality with up to 200 nodes.

Keywords: Vehicle routing problem, prize-collecting, multiobjective.

1 Introduction

Much effort has been made to increase the oil production in Brazil though the use of new technologies. As a consequence, the Brazilian oil production has met the country's need in 2006 and the country is globally self sufficient. The Rio Grande do Norte basin has been exploited for the last 30 years and about 98% of the oil wells are pumped using artificial lift systems. One such system is the Mobile Oil Recovery (MOR) unit. It consists of a truck equipped with a pumping system. The unit starts its tour at the depot, then it pumps several wells before returning to the depot at the end of the day. Whenever the unit's tank is full, an auxiliary vehicle is used to transfer the oil from the MOR unit to its own tank and to carry it to the depot. Thus, the MOR unit capacity can be considered unlimited.

The MOR optimization Problem (MORP) is a multiobjective problem which consists in finding a set of wells to be pumped in a working day to maximize the oil extraction and to minimize the travel time. The two objectives are opposite, one pushing to increase profit and the other to reduce costs. With one MOR unit, the problem is close to the Selective Traveling Salesman Problem which is also called Orienteering Problem or Maximum Collection Problem [8]. With a fleet of vehicles, the problem becomes a Vehicle Routing Problem (VRP) close to the Prize-Collecting VRP [2]. For further investigation on routing problems, readers are referred to the following works: the state of the art on exact and

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 283–292, 2008.

approximated methods for the VRP and its variants are found in [14] and an overview covering about 500 papers on classical routing problems are found in [9]. For multiobjective solutions strategies on routing problems, see [3,7].

A mathematical formulation for the MORP is proposed in [13] for a single MOR unit. Heuristics applications of the MORP are presented in [1,13]. We propose several formulations for the MORP with a single vehicle or a fleet of vehicles. They are strengthened by improving the subtour elimination constraints. Instances with up to 200 nodes, close to reality, are solved.

The paper is organized as follows: the problem definition and formulations for one unit are presented in Section 2. Sections 3 and 4 are devoted to the MORP with several units. Computational results are shown in Section 5 and final remarks are made in Section 6.

2 Formulations Using One Vehicle

The geographical data (roads, wells and depot) are modeled as an undirected graph $G = (N, E)$. G is preprocessed to build a complete digraph $G' = (V, A)$ where V is the set of wells and the depot v_0. Let d_{ij} be the shortest distance from i to j, $\forall (i,j) \in A$, and let s be the MOR unit average speed. Thus, for every arc of G', the travel time t_{ij} is computed as $t_{ij} = d_{ij}/s$.

Let t'_i be the total operation time at well i (time to connect the unit, to pump, and to disconnect the unit) and let p_i be its oil production. Let P and T be respectively the total oil production and the total working time of a MOR unit. Moreover, $\overline{T}$ is the maximal time an MOR unit can work in a day. Wells can be exploited only once a day as in the previous works [1,13].

Given K, the total number of MOR units, the MORP consists in defining one circuit $\tau = \{v_0, v_{\sigma 1}, v_{\sigma 2}, ..., v_{\sigma k}, v_0\}$ for each MOR unit, where σ is the position of wells in the circuit to be exploited in a day, such that P is maximized and T is minimized. The time limit $T \leq \overline{T}$ has to be satisfied.

As far as we know, only one formulation has been proposed in the literature for the MORP [13]. It considers one vehicle and the optimization is done in two phases: first, the maximal amount of oil is computed, and second, the shortest route to extract this amount is computed. In this section, our contributions improve the formulation proposed in [13] as follows: (i) remove the constraint ensuring the MOR unit returns to the depot because it is redundant, (ii) simplify the flow conservation constraints, (iii) test different strategies to eliminate invalid subtours, and (iv) strengthen the subtour elimination constraints.

Let $f_{ij} \in \{0, 1\}$ be the decision variable on the choice of arc (i, j) and let x_i be the binary variables which specify if well i is exploited or not. The first optimization phase for the MORP is given as follows:

$$\max \; P = \sum_{i \in V \setminus \{v_0\}} p_i \cdot x_i \qquad s.t. \tag{1}$$

$$\sum_{i \in V \setminus \{v_0\}} t'_i \cdot x_i + \sum_{(i,j) \in A} t_{ij} \cdot f_{ij} \leq \overline{T} \tag{2}$$

$$\sum_{j:(j,i)\in A} f_{ji} - \sum_{j:(i,j)\in A} f_{ij} = 0 \quad \forall i \in V\backslash\{v_0\} \tag{3}$$

$$\sum_{j:(j,i)\in A} f_{ji} = x_i \quad \forall i \in V\backslash\{v_0\} \tag{4}$$

$$\sum_{j\in V} f_{0j} = 1 \tag{5}$$

(subtour eliminations constraints) (6)

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{7}$$

$$f_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{8}$$

The objective function (1) aims at minimizing the oil extraction. Inequality (2) limits the working time (travel and operation time) to $\overline{T}$. The flow conservation constraints are (3) and (4). Restriction (5) guarantees the tour starts at the depot. Variables x_i and f_{ij} are respectively defined in Constraints (7) and (8). We discuss in Section 2.1 the use of several subtour elimination constraints: those of Miller and Tucker and Zemlin (MTZ) [5,11], and those of Gavish and Graves using either aggregated (GGA) or disaggregated flow (GGD) [6]. GGA constraints are used in [13].

The objective of the second optimization phase is to minimize the working time (9) subject to Constraints (3)–(8) and (10). Constraint (10) restricts the total production to be equal to the total optimal prize P^* obtained in the first phase.

$$\min \; T = \sum_{i\in V\backslash\{v_0\}} t'_i \cdot x_i + \sum_{(i,j)\in A} t_{ij} \cdot f_{ij} \qquad s.t. \tag{9}$$

$$\sum_{i\in V\backslash\{v_0\}} p_i \cdot x_i = P^* \tag{10}$$

Constraints (3)–(8).

2.1 Subtour Eliminations Constraints

A subtour is defined by any ordered subset of vertices. For the MORP, only subtours starting and ending at the depot v_0 are valid. Subtour constraints have been evaluated in the literature for the TSP problem, see e.g. [15]. MTZ, GGA and GGD subtour elimination constraints for the MORP, and some improvements are described below.

An upper bound on the number M of wells that can be exploited in a working day can be computed. Considering the working time of the MOR unit, a simple

procedure consists in computing M by sorting the wells in increasing order of operation time t'_i [13]. Thus, M is such that:

$$\sum_{i=1}^{M} t'_i \leq \overline{T} \leq \sum_{i=1}^{M+1} t'_i. \tag{11}$$

We propose to strengthen the value of M by using also the minimum travel time to arrive at each node. Moreover, the vehicle must return to the depot and the minimal time to return to the depot is also considered. M is given as:

$$\sum_{i=1}^{M} \left(t'_i + \min_{j \in V} \{t_{ji}\} \right) \leq \overline{T} - \min_{j \in V} \{t_{j0}\} \leq \sum_{i=1}^{M+1} \left(t'_i + \min_{j \in V} \{t_{ji}\} \right) \tag{12}$$

Lifted Miller and Tucker and Zemlin Constraints. The Miller, Tucker and Zemlin constraints define a topological order to eliminate invalid subtours. Variables u_i state the order well i appears in the tour. However, for the MORP, the depot appears twice (at the beginning and at the end). Thus, one can duplicate the depot and work on a support graph. We consider instead the depot only at the beginning of the topological design. This can be done since the flow structure defined by variables f_{ij} and x_i, and Constraints (3)–(5) guarantees the return to the depot. The corresponding MTZ constraints for the MORP is given in Equations (13)–(14).

$$u_i - u_j + M \cdot f_{ij} \leq M - 1 \qquad \forall (i,j) \in A, j \neq \{v_0\} \tag{13}$$

$$0 \leq u_i \leq M \qquad \forall i \in V \tag{14}$$

There is $O(|V|^2)$ of such constraints, improved by M. They can be lifted using the same ideas as Desrochers and Laporte [5]. It consists in adding a valid non-negative term $\alpha_{ji} f_{ji}$ to the Inequalities (13): $u_i - u_j + M \cdot f_{ij} + \alpha_{ji} \cdot f_{ji} \leq M - 1$. If $f_{ji} = 0$, then α_{ji} may take any value. Suppose now $f_{ji} = 1$. Then, the MOR unit exploits well $j \neq v_0$ before well i, $u_i = u_j + 1$. Thus, $f_{ji} = 1$ implies $f_{ij} = 0$ due to Constraints (3) and (4). By substitution, we obtain $\alpha_{ji} \leq M - 2$. The larger α_{ji}, the stronger is the lift. Thus, $\alpha_{ji} = M - 2$. A lifted version of Constraints (13) is given in Inequalities (15).

$$u_i - u_j + M \cdot f_{ij} + (M-2) \cdot f_{ji} \leq M - 1 \quad \forall (i,j) \in A, j \neq v_0 \tag{15}$$

Gavish and Graves Constraints. The Gavish and Graves [6] approach removes invalid subtours by building a network flow. A flow is sent to the nodes of the tour. Each node consumes one unit. In disaggregated flow, a specific flow is sent from the source to each node [4,10]. Otherwise, if the flow is not specified, it is an aggregated flow.

Let y_{ij} be the flow variable on arc (i,j). Thus, GGA constraints for the MORP are given in Equations (16)–(18). Constraints (16) are the flow conservation

constraints. Inequalities (17) state a flow uses the arc (i, j) if it is selected. These constraints are strengthened by using M. In this strategy, there are $O(|V|^2)$ constraints and variables.

$$\sum_{j:(j,i)\in A} y_{ji} - \sum_{j:(i,j)\in A} y_{ij} = x_i \quad \forall i \in V\backslash\{v_0\} \tag{16}$$

$$y_{ij} \leq M \cdot f_{ij} \qquad \forall (i,j) \in A, \tag{17}$$

$$y_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{18}$$

The GGD version is given in Constraints (19)–(22). Let y_{ij}^k be the variable specifying if flow for node k traverses arc (i, j) or not. Constraints (19) are the flow conservation constraints. Equations (20) state that a flow unit is sent from the source to each node k. Restrictions (21) specify that flow for node k traverses arc (i, j) if and only if it is used. This strategy implies $O(|V|^3)$ constraints and variables. Usually, it produces a better linear relaxation than using the aggregated flow.

$$\sum_{j:(i,j)\in A} y_{ij}^k - \sum_{j:(j,i)\in A} y_{ji}^k = 0 \qquad \forall k \in V\backslash\{v_0\}, \forall i \in V\backslash\{v_0, k\} \tag{19}$$

$$\sum_{j:(0,j)\in A} y_{0j}^k = x_k \qquad \forall k \in V\backslash\{v_0\} \tag{20}$$

$$y_{ij}^k \leq f_{ij} \qquad \forall k \in V\backslash\{v_0\}, \forall (i,j) \in A \tag{21}$$

$$y_{ij}^k \geq 0 \qquad \forall k \in V\backslash\{v_0\}, \forall (i,j) \in A \tag{22}$$

3 A Three-Indexed Formulation Using Several Vehicles

Let x_i^k be a decision variable that specifies if well i is exploited by the vehicle k or not. Variables $f_{ij}^k \in \{0, 1\}$ state if vehicle k exploits well j after well i or not. $P(K)$ is the total profit collected using the K MOR units. All other terms are defined in Section 2. The three-indexed formulation is as follows:

$$\max \; P(K) = \sum_{i\in V\backslash\{v_0\}} p_i \cdot \sum_{k=1}^{K} x_i^k \qquad s.t. \tag{23}$$

$$\sum_{i\in V\backslash\{v_0\}} t'_i \cdot x_i^k + \sum_{(i,j)\in A} t_{ij} \cdot f_{ij}^k \leq \overline{T} \qquad \forall k = 1, ..., K \tag{24}$$

$$\sum_{j:(j,i)\in A} f_{ji}^k - \sum_{j:(i,j)\in A} f_{ij}^k = 0 \qquad \forall k = 1, ..., K, \forall i \in V\backslash\{v_0, k\} \tag{25}$$

$$\sum_{j:(0,j)\in A} f_{0j}^k \leq 1 \qquad \forall k = 1, ..., K \tag{26}$$

$$\sum_{j:(j,i)\in A} f_{ji}^k = x_i^k \qquad \forall k = 1, ..., K,\ \forall i \in V\backslash\{v_0\} \tag{27}$$

$$\sum_{k=1}^{K} x_i^k \leq 1 \qquad \forall i \in V\backslash\{v_0\} \tag{28}$$

$$\sum_{j:(j,i)\in A} y_{ji} - \sum_{j:(i,j)\in A} y_{ij} = \sum_{k=1}^{K} x_i^k \qquad \forall i \in V\backslash\{v_0\} \tag{29}$$

$$y_{ij} \leq M \cdot \sum_{k=1}^{K} f_{ij}^k \qquad \forall (i,j) \in A, j \neq v_0 \tag{30}$$

$$y_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{31}$$

$$x_i^k \in \{0,1\} \qquad \forall k = 1, ..., K, \forall i \in V\backslash\{v_0\} \tag{32}$$

$$f_{ij}^k \in \{0,1\} \qquad \forall k = 1, ..., K, \forall (i,j) \in A \tag{33}$$

Restrictions (24) limit the units work in a day. The flow conservation constraints are defined in (25) and (26). Constraints (27) ensure that unit k pass though an arc (i, j) only if it exploits well j. Inequalities (28) specify that at most one unit visits a well in a day. Constraints (29) and (30) are the GGA subtour elimination constraints. Constraints (31)–(33) are the variables definition. This formulation contains $O(|V^3|)$ variables and constraints. The GGA constraints are chosen according to the computational results for one vehicle (Section 5). Obviously, other strategies could be used as well.

4 A Two-Indexed Formulation Using Several Vehicles

We do not explicitly define which unit exploits well i as every unit has the same characteristics (homogeneous fleet). A similar idea was previously used, for example, in [12]. Variables f_{ij} and x_i are defined in Section 2. Additionally, variables d_i specify the date (time) well i is visited by a vehicle in a day. The two-indexed formulation is given as follows:

$$\max\ P = \sum_{i\in V\backslash\{v_0\}} p_i \cdot x_i \qquad s.t. \tag{34}$$

$$\sum_{j:(j,i)\in A} f_{ji} - \sum_{j:(i,j)\in A} f_{ij} = 0 \qquad \forall i \in V\backslash\{v_0\} \tag{35}$$

$$\sum_{j:(j,i)\in A} f_{ji} = x_i \qquad \forall i \in V\backslash\{v_0\} \tag{36}$$

$$\sum_{j\in V} f_{0j} = K \tag{37}$$

$$d_i - d_j + (\overline{T} + t'_i + t_{ij}) \cdot f_{ij} + (\overline{T} - t'_j - t_{ji}) \cdot f_{ji} \leq \overline{T} \qquad \forall (i,j) \in A, i, j \neq v_0 \tag{38}$$

$$d_i \geq t_{0i} \cdot f_{0i} + \sum_{j\neq v_0} (t_{0j} + t'_j + t_{ji}) \cdot f_{ji} \qquad \forall i \in V\backslash\{v_0\} \tag{39}$$

$$d_i \leq \overline{T} - (t'_i + t_{i0}) \cdot f_{i0} - \sum_{j\neq v_0} (t'_i + t_{ij} + t'_j + t_{j0}) \cdot f_{ij} \qquad \forall i \in V\backslash\{v_0\} \tag{40}$$

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{41}$$

$$f_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{42}$$

$$d_i \geq 0 \qquad \forall i \in V\backslash\{v_0\} \tag{43}$$

The flow conservation is given in Constraints (35). Restrictions (36) ensure arc (i,j) is used if well j is exploited. Inequalities (37) state K MOR units are used. Constraints (38) link the time node j is visited, to the time node i is visited, and to the selection of arc (i,j). This is an adaptation of the lifted MTZ constraints (see Section 2.1). Inequalities (39) and (40) define generalized lower and upper bounds on the time node i is visited. Inequalities (39) link the time node i is visited to variables f_{ji}. At most one of the arcs entering node i is used. Thus, d_i is at least equal to the minimal time required to arrive at node i, either by going from v_0 to i or by going from j to i. The same idea applies to the Inequalities (40). Variables x_i^k, f_{ij}^k and d_i are defined respectively by Constraints (41) to (43). The two-indexed formulation has $O(|V|^2)$ variables and constraints. MTZ is used since the time constraints definition is straightforward and the formulation has still $O(|V|^2)$ variables.

5 Computational Results

The computational experiments were carried out on an Intel Core 2 Duo with 2.66 GHz clock and 4Gb of RAM memory, using CPLEX 11 under default parameters. Instances were generated using a geographical information system to simulate real situations. Comparison among the proposed formulations are measured in terms of time to prove optimality and of linear relaxation.

In the Tables 1 and 2, each line corresponds to an instance. For each instance, the working day length (L) in minutes, the number of wells ($|V|$) and its optimal production (P^*) are given. For each formulation, the linear relaxation

value (RL^*), the time (T) spent by the unit in the optimal solution, the time (time(s)) required to prove optimality in seconds (rounded up), and the number of nodes (nodes) explored in the branch-and-bound tree are presented. The symbol ($-$) means the solver did not prove optimality because it ran out of memory. When the optimal solution is unknown, the best integer solution found so far is identified by "($\geq$ value)".

Table 1 summarizes the results for the formulations for one vehicle using the MTZ, GGA or GGD subtour elimination constraints. From the computational results, GGA proves optimality faster than MTZ and GGD for 9 instances. MTZ proves optimality faster than GGA and GGD for 7 instances. In spite of having the worst linear relaxation, MTZ is able to prove optimality for instances with up to 200 nodes. GGD consumes a lot of time even if it produces good linear relaxation. An interesting result on the linear relaxation is found for $L = 480$ and $|V| = 20$: the GGA linear relaxation is better than the linear relaxation of GGD. This happens here because the value of M is equal to the optimal amount of wells exploited in a day.

We have also run the second optimization phase for all instances presented in Table 1. The time T was only improved for the instance with $L = 480$ and $|V| = 120$ ($T^* = 479.5$ instead of $T = 480$). Thus, results of the second optimization phase were not tested for several vehicles. Even so, it remains valuable since it takes place in the global decision process of the problem.

The results for the two-indexed and the three-indexed formulations are presented in Table 2. The number of vehicles used (K) and the sum of the total time spent by all the MOR units (T') are given. The three-indexed formulation produces a better linear relaxation than the two-indexed formulation. However,

Table 1. The first optimization phase for the MORP using one vehicle

				MTZ			GGA				GGD			
L	$\|V\|$	P^*	RL^*	T	time	nodes	RL^*	T	time	nodes	RL^*	T	time	nodes
480	20	12.60	17.21	477	2	1292	12.92	477	0	1	13.41	477	11	17
480	30	15.88	18.38	477	5	2337	17.19	477	6	979	16.61	477	841	191
480	40	15.88	18.38	477	7	2081	17.19	477	6	963	16.61	477	30137	255
480	60	15.84	18.43	467	15	3203	16.86	467	3	90	-	-	-	-
480	80	9.97	13.80	479	83	5429	11.67	479	32	1410	-	-	-	-
480	120	18.73	19.09	480	73	2996	19.05	480	204	1910	-	-	-	-
480	160	19.10	19.47	480	240	2095	19.46	480	2540	3378	-	-	-	-
480	200	19.64	19.82	480	62	3256	19.77	480	20626	2360	-	-	-	-
960	20	24.45	32.11	952	817	915570	28.39	952	98	14807	25.65	960	452	264
960	30	31.65	35.93	950	424	228949	35.29	950	446	46898	32.43	950	5313	527
960	40	19.76	24.17	909	406	40244	23.84	909	135	10898	22.34	909	63631	2401
960	60	31.65	35.96	950	974	301668	35.18	950	377	33257	32.26	-	-	-
960	80	37.70	38.05	959.5	3240	57320	38.00	959.5	866	21595	-	-	-	-
960	120	37.99	38.41	960	2764	36803	38.42	960	872	6716	-	-	-	-
960	160	40.05	40.19	960	377	5893	40.16	960	585	877	-	-	-	-
960	200	40.05	40.19	960	420	6672	40.16	960	789	951	-	-	-	-

Table 2. The first optimization phase for the MORP using several vehicles

				Two-indexed formulation				Three-indexed formulation			
K	L	$\|V\|$	P^*	RL*	T'	time (s)	nodes	RL*	T'	time (s)	nodes
2	480	10	20.97	27.74	911	2	8416	24.09	**870**	22	9139
2	480	20	24.45	29.55	953	74	41047	25.08	953	4	487
2	480	30	31.16	35.79	941	51	51661	33.78	941	1149	75216
2	480	40	31.16	35.79	941	646	84815	33.78	941	1315	64397
2	480	50	28.99	34.70	928	654	94922	30.93	928	7929	101935
2	480	60	30.39	35.78	946	326	57405	32.99	946	1619	60724
2	480	70	25.88	32.86	916	2	3374	30.43	916	1219	26108
2	480	80	19.35	27.03	881	78	69927	22.86	**876**	16426	313421
2	960	20	$\geq$ 46.51	55.89	-	-	-	52.75	-	-	-
2	960	30	$\geq$ 62.26	69.44	-	-	-	68.57	-	-	-
3	480	10	29.82	29.82	909	1	503	29.82	**901**	2	639
3	480	20	33.72	40.42	943	553	704266	36.73	943	27788	214856
3	480	30	45.49	52.31	943	2394	2269664	49.86	-	-	-
3	960	20	62.16	62.16	933	134	138791	62.16	-	-	-
3	960	30	$\geq$ 88.78	98.44	-	-	-	97.63	-	-	-

the two-indexed formulation performs better to compute the optimal solution. In addition to the number of wells, the problem becomes more difficult when the number of vehicles increases. Moreover, the working day limit also contributes to the difficulty of the problem. Results suggest it is suitable to use a small time window (480 minutes). The three-indexed formulation found sometimes a smaller value of T' as shown in bold.

6 Concluding Remarks

Several formulations for the MORP are proposed in this work and the first ever results using several vehicles are presented. Additionally, we proposed to improve the subtour constraints by taking advantage of the time window. Thus, instances close to reality (up to 200 wells) are solved. Among the formulations for one vehicle, GGA performs globally better than MTZ and GGD to prove optimality. For several vehicles, the two-indexed formulation is faster to prove optimality in spite of weaker linear relaxations.

Computational experiments show that the time window restriction plays a key role in computing an optimal solution: the smaller the time window, the easier the problem to solve. Optimal solutions can be computed for medium-sized instances with two MOR units. When using three vehicles, this does not hold as the CPU time increases dramatically for small instances.

The larger instances used here are larger than the problems considered by the company in the Rio Grande do Norte Basin. Consequently, the oil company is now able to compute the optimal solution for the MORP instead of using solutions given by heuristics. It could be interesting in future work to investigate

instances characteristics to specify situations where the second optimization phase becomes really useful. Moreover, for large time windows, we could investigate an approach to split it.

Acknowledgments. We thank Petrobras staff for providing valuable informations about the MOR unit and its usage in a petrol field.

References

1. Aloise, D., Aloise, D.J., Ochi, L.S., Maia, R.S., Bittencourt, V.G.: Uma colônia de formigas para o problema de explotação de petróleo e otimização de rotas de unidades móveis de pistoneio. In: Congresso Brasileiro de Automática, Natal., pp. 1232–1237 (2002)
2. Balas, E.: The prize collecting traveling salesman problem. Networks 19, 621–636 (1989)
3. Boffey, B.: Multiobjective routing problems. TOP 3, 167–220 (1995)
4. Corte-Real, M., Gouveia, L.: Network flow models for the local access network expansion problem. Computers and Operations Research 34, 1141–1157 (2007)
5. Desrochers, M., Laporte, G.: Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. Operations Research Letters 10, 27–36 (1991)
6. Gavish, B., Graves, S.C.: The travelling salesman problem and related problems. Technical Report OR 078-78, Massachusetts Institute of Technology (2005)
7. Keller, C.K., Goodchild, M.F.: The multiobjective vending problem: a generalization of the travelling salesman problem. Environment and Planning B: Planning and Design 15, 447–460 (1988)
8. Laporte, G., Martelo, S.: The selective travelling salesman problem. Discrete Applied Mathematics 26, 193–207 (1990)
9. Laporte, G., Osman, I.H.: Routing problems: A bibliography. Annals of Operations Research 61, 227–262 (1995)
10. Magnanti, T.L., Wolsey, L.: Optimal trees in network models. In: Handbooks in operations research and management science, vol. 7, pp. 503–615. Elsevier, North-Holland (1995)
11. Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulations and traveling salesman problems. Journal of the ACM 7, 326–329 (1960)
12. Ropke, S., Cordeau, J.F., Laporte, G.: Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks 49, 258–272 (2007)
13. Santos, A.C., Barros, C.A., Aloise, D.J., Neves, J.A., Noronha, T.F.: Um algoritmo GRASP reativo aplicado ao problema do emprego da unidade móvel de pistoneio. In: XXXIII Simpósio Brasileiro de Pesquisa Operacional, Campos do Jordão, pp. 247–258 (2001)
14. Toth, P., Vigo, D.: The vehicle routing problem. Society for Industrial & Applied Mathematics. SIAM, Philadelphia (2002)
15. Wong, R.T.: Integer programming formulations and traveling salesman problems. In: Proceedings IEEE Conference on Circuits and Computers, pp. 149–152. IEEE Press, Los Alamitos (1980)

Empirical Analysis of an Online Algorithm for Multiple Trading Problems

Esther Mohr[1] and Günter Schmidt[1,2,⋆]

[1] Saarland University, P.O. Box 151150, D-66041 Saarbrücken, Germany
[2] University of Liechtenstein, Fürst-Franz-Josef-Strasse, 9490 Vaduz, Liechtenstein
em@itm.uni-sb.de, gs@itm.uni-sb.de

Abstract. If we trade in financial markets we are interested in buying at low and selling at high prices. We suggest an active trading algorithm which tries to solve this type of problem. The algorithm is based on reservation prices. The effectiveness of the algorithm is analyzed from a worst case and an average case point of view. We want to give an answer to the questions if the suggested active trading algorithm shows a superior behaviour to buy-and-hold policies. We also calculate the average competitive performance of our algorithm using simulation on historical data.

Keywords: online algorithms, average case analysis, stock trading, trading rules, performance analysis, competitive analysis, trading problem, empirical analysis.

1 Introduction

Many major stock markets are electronic market places where trading is carried out automatically. Trading policies which have the potential to operate without human interaction are of great importance in electronic stock markets. Very often such policies are based on data from technical analysis [8], [6], [7]. Many researchers have also studied trading policies from the perspective of artificial intelligence, software agents and neural networks [1], [5], [9].

In order to carry out trading policies automatically they have to be converted into trading algorithms. Before a trading algorithm is applied one might be interested in its performance. The performance analysis of trading algorithms can basically be carried by three different approaches. One is Bayesian analysis where a given probability distribution for asset prices is a basic assumption. Another one is assuming uncertainty about asset prices and analyzing the trading algorithm under worst case outcomes; this approach is called competitive analysis. The third one is a heuristic approach where trading algorithms are designed and the analysis is done on historic data by simulation runs. In this paper we apply the second and the third approach in combination. We consider a multiple trade problem and analyze an appropriate trading algorithm from a worst case

⋆ Corresponding author.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 293–302, 2008.

point of view. Moreover we evaluate its average case performance empirically and compare it to other trading algorithms.

The reminder of this paper is organized as follows. In the next section the problem is formulated and a worst case competitive analysis of the proposed trading algorithm is performed. In Section 3 different trading policies for the multiple trade problem are introduced. Section 4 presents detailed experimental findings from our simulation runs. We finish with some conclusions in the last section.

2 Problem Formulation

If we trade in financial markets we are interested in buying at low prices and selling at high prices. Let us consider the single trade and the multiple trade problem. In a *single trade problem* we search for the minimum price m and the maximum price M in a time series of prices for a single asset. At best we buy at price m and sell later at price M. In a *multiple trade problem* we trade assets sequentially in a row, e.g. we buy some asset u today and sell it later in the future. After selling asset u we buy some other asset v and sell it later again; after selling v we can buy w which we sell again, etc. If we buy and sell (trade) assets k times we call the problem k-trade problem with $k \geq 1$.

As we do not know future prices the decisions to be taken are subject to uncertainty. How to handle uncertainty for trading problems is discussed in [3]. In [2] and [4] online algorithms are applied to a search problem. Here a trader owns some asset at time $t = 0$ and obtains a price quotation $m \leq p(t) \leq M$ at points of time $t = 1, 2, \ldots, T$. The trader must decide at every time t whether or not to accept this price for selling. Once some price $p(t)$ is accepted trading is closed and the trader's payoff is calculated. The horizon T and the possible minimum and maximum prices m and M are known to the trader. If the trader did not accept a price at the first $T - 1$ points of time he must be prepared to accept some minimum price m at time T. The problem is solved by an online algorithm.

An algorithm ON computes online if for each $j = 1, \ldots, n-1$, it computes an output for j before the input for $j + 1$ is given. An algorithm computes offline if it computes a feasible output given the entire input sequence $j = 1, \ldots, n - 1$. We denote an optimal offline algorithm by OPT. An online algorithm ON is c-competitive if for any input I

$$ON(I) > 1/c * OPT(I). \tag{1}$$

The competitive ratio is a worst-case performance measure. In other words, any c-competitive online algorithm is guaranteed a value of at least the fraction $1/c$ of the optimal offline value $OPT(I)$, no matter how unfortunate or uncertain the future will be. When we have a maximization problem $c \geq 1$, i.e. the smaller c the more effective is ON. For the search problem the policy (trading rule) [2]

accept the first price greater or equal to reservation price $p* = \sqrt{(M * m)}$

has a competitive ratio $c_s = \sqrt{\frac{M}{m}}$ where M and m are upper and lower bounds of prices $p(t)$ with $p(t)$ from $[m, M]$. c_s measures the worst case in terms of maximum and minimum price.

This result can be transferred to k-trade problems if we modify the policy to

buy the asset at the first price smaller or equal and sell the asset at the first price greater or equal to reservation price $p* = \sqrt{(M * m)}$.

In the single trade problem we have to carry out the search twice. In the worst case we get a competitive ratio of c_s for buying and the same competitive ratio of c_s for selling resulting in an overall competitive ratio for the single trade problem of $c_t = c_s c_s = M/m$. In general we get for the k-trade problem a competitive ratio of $c_t(k) = \prod_{i=1,\ldots,k} (M(i)/m(i))$. If m and M are constant for all trades $c_t(k) = (M/m)^k$. The ratio c_t can be interpreted as the rate of return we can achieve by buying and selling assets.

The bound is tight for arbitrary k. Let us assume for each of k trades we have to consider the time series $(M, (M * m)^{1/2}, m, m, (M * m)^{1/2}, M)$. OPT always buys at price m and sells at price M resulting in a return rate of M/m; ON buys at price $(M * m)^{1/2}$ and sells at price $(M * m)^{1/2}$ resulting in a return rate of 1, i.e. $OPT/ON = M/m = c$. If we have k trades OPT will have a return of $(M/m)^k$ and ON of 1^k , i.e. $OPT(k)/ON(k) = (M/m)^k = c(k)$.

In the following we apply the above modified reservation price policy to multiple trade problems.

3 Multiple Trade Problem

In a multiple trade problem we have to choose points of time for selling current assets and buying new assets over a known time horizon. The horizon consists of several trading periods i of different types p; each trading period consists of a constant number of h days. We differ between $p = 1, 2, \ldots, 6$ types of periods with length h from $\{7, 14, 28, 91, 182, 364\}$ days e.g. period type $p = 6$ has length $h = 364$ days; periods of type p are numbered with $i = 1, \ldots, n(p)$. There is a fixed length h for each period type p, e.g. period length $h = 7$ corresponds to period type $p = 1$, period length $h = 14$ corresponds to period type $p = 2$, etc. For a time horizon of one year, for period type $p = 1$ we get $n(1) = 52$ periods of length $h = 7$, for type $p = 2$ we get $n(2) = 26$ periods of length $h = 14$, etc.

We may choose between three trading policies. Two elementary ones are Buy-and-Hold $(B + H)$, a passive policy, and Market Timing (MT), an active policy. The third one is a random (Rand) policy. As a benchmark we use an optimal offline algorithm called Market (MA). We assume that for each period i there is an estimate of the maximum price $M(i)$ and the minimum price $m(i)$. Within each period $i = 1, \ldots, n(p)$ we have to buy and sell an asset at least once. The annualized return rate $R(x)$, with x from $\{MT, \text{Rand}, B + H, MA\}$ is the

performance measure used. At any point of time of the horizon the policy either holds an asset or an overnight deposit.

In order to describe the different policies we define a holding period with respect to MT. A holding period is the number of days h between the purchase of asset j and the purchase of another asset j' ($j' \neq j$) by MT. Holding periods are determined by either reservation prices $RP_j(t)$ which give a trading signal or when the last day T of the period is reached.

MARKET TIMING (MT)

MT calculates reservation prices $RP_j(t)$ for each day t for each asset j. At each day t, MT must decide whether to sell asset j or to hold it another day considering the reservation prices. Each period i, the first offered price $p_j(t)$ of asset j with $p_j(t) \geq RP_j(t)$ is accepted by MT and asset j is sold. The asset $j*$, which is bought by MT is called MTasset. MT chooses the MTasset $j*$ if $RP_{j*}(t) - p_{j*}(t) = \max\{RP_j(t) - p_j(t) | j = 1, \ldots, m\}$ and $p_{j*}(t) < RP_{j*}(t)$. If there was no trading signal in a period related to reservation prices then trading is done on the last day T of a period. In this case MT must sell asset j and invest in asset j' at day T. The holding period of MT showing buying (Buy) and selling ($Sell$) points and intervals with overnight deposit (OD) is shown in Fig. 1.

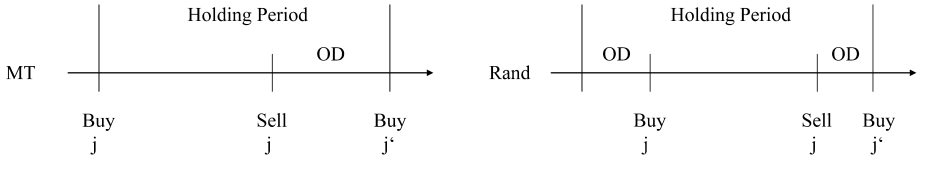


Fig. 1. Holding period for MT and for Rand

RANDOM (Rand)

Rand will buy and sell at randomly chosen prices $p_j(t)$ within the holding period of MT (cf. Fig. 1).

BUY AND HOLD ($B + H$)

$B + H$ will buy at the first day t of the period and sell at the last day T of the period.

MARKET (MA)

To evaluate the performance of these three policies empirically we use as a benchmark the optimal offline policy. It is assumed that MA knows all prices $p_j(t)$ of a period including also these which were not presented to MT if there were any. In each period i MA will buy at the minimum price $p_{min} > m(i)$ and sell

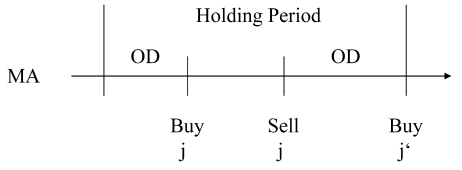


Fig. 2. Holding period for MA

at the maximum possible price $p_{max} < M(i)$ within the holding period of MT (cf. Fig. 2).

The performance of the investment policies is evaluated empirically. Clearly, all policies cannot beat the benchmark policy MA.

4 Experimental Results

We want to investigate the performance of the trading policies discussed in Section 3 using experimental analysis. Tests are run for all $p = 1, 2, \ldots, 6$ period types with the number of periods $n(p)$ from $\{52, 26, 13, 4, 2, 1\}$ and period length h from $\{7, 14, 28, 91, 182, 364\}$ days. The following assumptions apply for all tested policies:

1. There is an initial portfolio value greater zero.
2. Buying and selling prices $p_j(t)$ of an asset j are the closing prices of day t.
3. At each point of time all money is invested either in assets or in 3% overnight deposit.
4. Transaction costs are 0.0048% of the market value but between 0.60 and 18.00 Euro.
5. When selling and buying is on different days the money is invested in overnight deposit.
6. At each point of time t there is at most one asset in the portfolio.
7. Each period i at least one buying and one selling transaction must be executed. At the latest on the last day of each period asset j has to be bought and on the last day it has to be sold.
8. In period $i = 1$ all policies buy the same asset j on the same day t at the same price $p_j(t)$; the asset chosen is the one MT will chose (MTasset).
9. In periods $i = 2, \ldots, n(p) - 1$ trades are carried out according to the different policies.
10. In the last period $i = n(p)$ the asset has to be sold at the last day of that period. No further transactions are carried out from there on.
11. If the reservation price is calculated over h days, the period length is (also) h days.

We simulate all policies using historical XETRA DAX data from the interval 2007.01.01 until 2007.12.31. This interval we divide into $n(p)$ periods where $n(p)$ is from $\{52, 26, 13, 4, 2, 1\}$ and p is from $\{7, 14, 28, 91, 182, 364\}$. With this arrangement we get 52 periods of length 7 days, 26 periods of length 14 days, etc. We carried out simulation runs in order to find out

(1) if MT shows a superior behaviour to buy-and-hold policies
(2) the influence of m and M on the performance of MT
(3) the average competitive ratio for policies for MA and MT.

Two types of buy-and-hold policies are used for simulation; one holds the MTasset within each period (MT_{B+H}) and the other holds the index over all periods (Index_{B+H}) of a simulation run. Thus, MT_{B+H} is synchronized with the MT policy, i.e, MT_{B+H} buys on the first day of each period the same asset which MT buys first in this period (possibly not on the first day) and sells this asset on the last day (note that this asset may differ from the one MT is selling on the last day) of the period. Using this setting we compare both policies related to the same period. Index_{B+H} is a common policy applied by ETF investment funds and it is also often used as a benchmark although it is not synchronized with the MT policy. In addition to these policies also the random policy Rand is simulated. Rand buys the same asset which MT buys on a randomly chosen day within a holding period.

We first concentrate on question (1) if MT shows a superior behaviour to the policies MT_{B+H} and Index_{B+H}. For calculating the reservation prices we use estimates from the past, i.e. in case of a period length of h days m and M are taken from the prices of these h days which are preceding the actual day t^* of the reservation price calculation, i.e. $m = \min\{p(t)|t = t^* - 1, t^* - 2, \ldots, t^* - h\}$ and $M = \max\{p(t)|t = t^* - 1, t^* - 2, \ldots, t^* - h\}$. In Table 1 the trading results are displayed considering also transaction costs. The return rates are calculated covering a time horizon of one year. For the three active policies (MA, MT, Rand) the transaction costs are the same because all follow the holding period of MT; in all these cases there is a flat minimum transaction fee.

Table 1. Annualized return rates for different period lengths

Historic	Annualized Returns Including Transaction Costs					
Policy	1 Week $n(7) = 52$	2 Weeks $n(14) = 26$	4 Weeks $n(28) = 13$	3 Months $n(91) = 4$	6 Months $n(182) = 2$	12 Months $n(364) = 1$
MA	418.18%	138.40%	201.61%	47.93%	72.95%	61.95%
MT	41.08%	1.37%	54.86%	6.08%	32.39%	31.35%
MT_{B+H}	9.70%	0.50%	17.18%	15.80%	45.30%	35.29%
Index_{B+H}	20.78%	20.78%	20.78%	20.78%	20.78%	20.78%
Rand	-23.59%	-21.23%	17.18%	-18.23%	6.20%	15.42%

MT dominates MT_{B+H} and Index_{B+H} in two cases (1 and 4 weeks). MT_{B+H} dominates MT and Index_{B+H} in two cases (6 and 12 months). Index_{B+H} dominates MT and MT_{B+H} in two cases (2 weeks and 3 months). MT generates the best overall annual return rate when applied to 4 weeks. MT_{B+H} generates the worst overall annual return rate when applied to 2 weeks. MT_{B+H} policy improves its performance in comparison to Index_{B+H} and MT policy proportional to the length of the periods. We might conclude the longer the period the

better the relative performance of MT_{B+H}. MT outperforms Index_{B+H} in four of six cases and it outperforms MT_{B+H} in three of six cases; MT and MT_{B+H} have the same relative performance. If the period length is not greater than 4 weeks MT outperforms MT_{B+H} in all cases. If the period length is greater than 4 weeks MT_{B+H} outperforms MT in all cases. Index_{B+H} outperforms MT_{B+H} in three of six cases. If we consider the average performance we have 27.86% for MT, 20.78% for Index_{B+H}, and 20.63% for MT_{B+H}. MT is not always the best but it is on average the best. From this we conclude that MT shows on average a superior behaviour to buy-and-hold policies under the assumption that m and M are calculated by historical data.

In general we would assume that the better the estimates of m and M the better the performance of MT. Results in Table 1 show, that the longer the periods the worse the relative performance of MT. This might be due to the fact that for longer periods historical m and M are worse estimates in comparison to those for shorter periods. In order to analyze the influence of estimates of m and M we run all simulations also with the observed m and M of the actual periods, i.e. we have optimal estimates. Results for optimal estimates are shown in Table 2 and have to be considered in comparison to the results for historic estimates shown in Table 1.

Now we can answer question (2) discussing the influence of m and M on the performance of MT. The results are displayed in Table 2. It turns out that in all cases the return rate of policy MT improves significantly when estimates of m and M are improved. For all period lengths now MT is always better than MT_{B+H} and Index_{B+H}. From this we conclude that the estimates of m and M are obviously of major importance for the performance of the MT policy. Now we concentrate on question (3) discussing the average competitive ratio for policies MA and MT. We now compare the experimental competitive ratio c_{ec} to the analytical competitive ratio c_{wc}. To do this we have to calculate OPT and ON for the experimental case and the worst case. We base our discussion on the return rate as the performance measure. We assume that we have precise forecasts for m and M.

A detailed example for the evaluation of the competitive ratio is presented in Table 3 considering a period length of 12 months. In this period six trades were executed using reservation prices based on the clairvoyant test set. The analytical results are based on the values of m and M for each holding period.

Table 2. Annualized returns for optimal historic estimates

Clairvoryant	Annualized Returns Including Transaction Costs					
Policy	1 Week $n(7)=52$	2 Weeks $n(14)=26$	4 Weeks $n(28)=13$	3 Months $n(91)=4$	6 Months $n(182)=2$	12 Months $n(364)=1$
MA	418.18%	315.81%	280.94%	183.43%	86.07%	70.94%
MT	102.60%	87.90%	76.10%	81.38%	55.11%	54.75%
MT_{B+H}	9.70%	-4.40%	22.31%	19.79%	45.30%	35.29%
Index_{B+H}	20.78%	20.78%	20.78%	20.78%	20.78%	20.78%
Rand	-23.59%	-101.3%	-10.67%	47.37%	46.08%	15.42%

Table 3. Periodic results for period length one year

Clairvoyant Data		Analytical Results			Experimental Results			
# Trades $n(364) = 1$	Holding Period	m	M	$c_{wc} = M/m$	Buy at MA/MT	Sell at	Periodic Return	$c_{ex} = MA/MT$
1^{st} trade	Week 1-14	37.91	43.23	1.1403				1.0072
MA					37.91	43.23	1.1403	
MT					37.91	42.92	1.1322	
2^{nd} trade	Week 14-24	34.25	38.15	1.1139				1.0069
MA					34.25	38.15	1.1139	
MT					34.25	37.89	1.1063	
3^{rd} trade	Week 24-25	13.54	13.69	1.0111				1.0000
MA					13.54	13.69	1.0111	
MT					13.54	13.69	1.0111	
4^{th} trade	Week 25-30	33.57	35.73	1.0643				1.0167
MA					33.57	35.73	1.0643	
MT					34.13	35.73	1.0469	
5^{th} trade	Week 30-46	51.23	58.86	1.1489				1.0646
MA					51.23	58.86	1.1489	
MT					52.37	56.52	1.0792	
5^{th} trade	Week 46-52	82.16	89.4	1.0881				1.0061
MA					82.16	89.4	1.0881	
MT					82.66	89.4	1.0815	

Table 4. Competitive ratio and annualized return rates

Clairvoyant Data		Analytical Results	Experimental Results			
Period Length	# Trades	OPT/ON	MA	MT	MA/MT	c_{ex}/c_{wc}
12 Months	6	1.7108	71.08%	54.89%	1.2950	75.69%
6 Months	7	1.8624	86.24%	55.28%	1.5601	83.77%
3 Months	18	2.8387	183.87%	81.82%	2.2473	79.16%
4 Weeks	38	3.8185	281.85%	77.02%	3.6594	95.83%
2 Weeks	48	4.1695	316.95%	89.05%	3.5592	85.36%
1 Week	52	4.1711	317.11%	103.84%	3.0538	73.21%

The analytical results are based on the consideration that MA achieves the best possible return and MT achieves a return of zero. E.g. for the first trade MA achieves a return rate of 14.03% and MT achieves a return rate of 0% i.e. MT achieves absolutely 14.03% less than MA and relatively a multiple of 1.1403. The experimental results are also based on the consideration that MA achieves the best possible return and MT now achieves the return rate generated during the experiment. E.g. for the first trade MA achieves a return rate of 1.1403

or 14.03% and MT achieves a return rate of 1.1322 or 13.22%. We compared the analytical results with the experimental results based on annualized return rates for the period lengths 1, 2, 4 weeks, 3, 6, and 12 months. The overall competitive ratio is based on period adjusted annual return rates. The results for all period lengths are presented in Table 4. Transaction costs are not taken into account in order not to bias results. As the policies are always invested there is no overnight deposit. E.g. For the period of 12 months the analytical worst case ratio OPT/ON is 1.7108 and the average experimental ratio MA/MT is 1.2950. The values of the competitive ratios for the other period lengths are also given in Table 4. The return of MT reached in the experiments reaches at least 27.33%, at most 77.22% and on average 45.67% of the return of MA.

5 Conclusions

In order to answer the three questions from section 4 twelve simulation runs were performed. MT outperforms buy-and-hold in all cases even when transaction costs are incorporated in the clairvoyant test set. Tests on historical estimates of m and M show that MT outperforms buy-and-hold in one third of the cases and also on average. We conclude that when the period length is small enough MT outperforms $B + H$.

It is obvious that the better the estimates of m and M the better the performance of MT. Results show that the shorter the periods, the better are estimates by historical m and M. As a result, the performance of MT gets worse the longer the periods become.

In real life it is very difficult to get close to the (analytical) worst cases. It turned out that the shorter the periods are the less MT achieves in comparison to MA. A MT trading policy which is applied to short periods leads to small intervals for estimating historical m and M. In these cases there is a tendency to buy too late (early) in increasing (decreasing) markets and to sell too late (early) in decreasing (increasing) markets due to unknown overall trend directions, e.g. weekly volatility leads to wrong selling decisions during an upward trend.

The paper leaves also some open questions for future research. One is that of better forecasts of future upper and lower bounds of asset prices to improve the performance of MT. The suitable period length for estimating m and M is an important factor to provide a good trading signal, e.g. if the period length is h days estimates for historical m and M were also be calculated over h days. Simulations with other period lengths for estimating m and M could be of interest. Moreover, the data set of one year is very small. Future research should consider intervals of 5, 10, and 15 years.

References

[1] Chavarnakul, T., Enke, D.: Intelligent technical analysis based equivolume charting for stock trading using neural networks. Expert Systems and Applications 34, 1004–1017 (2008)

[2] El-Yaniv, R.: Competitive solutions for online financial problems. ACM Computing Surveys 30, 28–69 (1998)
[3] El-Yaniv, R., Fiat, A., Karp, R., Turpin, G.: Optimal search and one-way trading algorithm. Algorithmica 30, 101–139 (2001)
[4] El-Yaniv, R., Fiat, A., Karp, R., Turpin, G.: Competitive analysis of financial games. In: IEEE Symposium on Foundations of Computer Science, pp. 327–333 (1992)
[5] Feng, Y., Ronggang, Y., Stone, P.: Two Stock Trading Agents: Market Making and Technical Analysis. In: Faratin, P., Parkes, D.C., Rodriguez-Aguilar, J.A., Walsh, W.E. (eds.) Agent Mediated Electronic Commerce V: Designing Mechanisms and Systems. LNCS (LNAI), pp. 18–36. Springer, Heidelberg (2004)
[6] Ratner, M., Leal, R.P.C.: Tests of technical trading strategies in the emerging equity markets of Latin America and Asia. Journal of Banking and Finance 23, 1887–1905 (1999)
[7] Ronggang, Y., Stone, P.: Performance Analysis of a Counter-intuitive Automated Stock Trading Strategy. In: Proceedings of the 5th International Conference on Electronic Commerce. ACM International Conference Proceeding Series, vol. 50, pp. 40–46 (2003)
[8] Shen, P.: Market-Timing Strategies that Worked. Working Paper RWP 02-01, Federal Reserve Bank of Kansas City, Research Division (May 2002)
[9] Silaghi, G.C., Robu, V.: An Agent Policy for Automated Stock Market Trading Combining Price and Order Book Information. In: ICSC Congress on Computational Intelligence Methods and Applications, pp. 4–7 (2005)

A Novel Approach for the Nurse Scheduling Problem

Serap Ulusam Seckiner

Gaziantep University
Faculty of Engineering
Department of Industrial Engineering
27310, Gaziantep, Turkey

Abstract. This study considers nurse scheduling problem in seven-days-three-week operations under an arrangement called 4-day workweek. A computer-assisted scheduling program has been proposed to schedule nurses with multiple shifts. The program has been developed in Delphi 7.0 and includes a scheduling module which schedules nurses for three weeks on the basis of the weekday and weekend shift nursing requirements. The program is significantly beneficial for scheduling of large numbers of nurses and providing optimal three weekly schedules in a reasonable time and provides a faster response, a reduced cost as compared to human experts and a number of practical features.

Keywords: Novel approach, Nurse scheduling, Flexible workweek.

1 Introduction

Many operations which have switched to a compressed workweek (CW) cite higher morale and productivity, decreases in turnover, absenteeism, overtime, requests for days off, tardiness, work commuting and easier recruitment [1]. CWs have gained much popularity in the workplaces. CW scheduling is crucial for efficient nursing management and a type of shift scheduling that concerns deals with matching weekday/weekend demand and resources, the resource being the nurses to be scheduled.

Much of the workforce scheduling literature is on single-shift operations under the conventional five-day workweek (see [2-6]). However, more and more seven-days-a-week operations are switching to novel workweeks such as three-day and four-day workweeks [1]. Recently, there has been a lot of interest in alternative work schedules, including 4-day workweeks. This research interest was illustrated by Alfares [6, 7], Narasinhan [8], Burns, Narasimhan, and Smith [9], Lankford [10], Hung [11], Hung and Emmons [12], Nanda and Browne [13]. There is workforce scheduling in a compressed workweek arrangements used in many 7-days-a-week operations: the 3-4 workweek (see Arnold and Mills [14], Poor [15], Steward and Larsen [16], Hung [17] Narasimhan [18] Billionnet [19]). However, these mentioned authors have scheduled personnel with either by hand or mathematical programming models. To find robust and acceptable solutions

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 303–307, 2008.

for the problems within an affordable time period is very important especially real world conditions. Many problems of service industry remain with difficulties to be solved within a reasonable time due to the complexity and dynamic nature of the service systems. Because of this reason, service systems such as hospital need more efficient and rapid computer-assisted scheduling programs.

The objective of this study is to describe and report the development process of the nurse scheduling program for compressed workweek schedules. The used algorithm allocates the workforce demands for days-on and days-off, assigns shifts to the schedule subject to demands and the shift change constraints and assigns off-days on the schedule subject to off-day and workstretch constraints. The advantage of the proposed computer program is to obtain feasible schedules easily and quickly from one week to the next week. If you run the program several times, the program gives different schedule. With the help of the program, the multiple-shift workforce scheduling model under the compressed workweek problem gets easily solved. Especially, the larger number of workforces, the harder solution is obtained. The proposed approach helps to find well-designed work schedules in a reasonable time.

2 Nurse Scheduling Environment

The algorithm provides optimal solution for the CW problem under these assumptions; A week runs from Sunday to Saturday. There are 3 shifts (Morning-Mid-Night) each day- Shifts may overlap. There must be at least D_j nurses on duty on shift j on a weekday and at least E_j nurses on duty on shift j on a weekend day. It is assumed that $D_j \geq \mathrm{E}_j$, j =1,2,3 ., Each nurse works only one shift per day. A nurse is said to be off on a day if the nurse does not work on any shift on that day. With the 4-day workweek, each nurse must work four days and receive three off-days each week. A nurse must receive at least one off-day before changing shifts. In addition, in a planning horizon of B weeks plus one day (i.e. Sunday of week 1 to Sunday of week *(B+1)*, each nurse must receive at least A out of the B weekends off. The objective is to minimize the workforce size subject to satisfying the above staffing requirements and the workrules. The program is first to find the smallest workforce size and then create an off-day and shift assignment in such a way that the resulting schedule will be feasible. The program calls Sat of week q and Sun of week (q +1) weekend q, (q =1.....,B).

The proposed program has been programmed in Borland Delphi 7.0 and if reader wants to see the code, it is supplied by author. Borland Delphi 7.0 is a complete environment for the visual design, compilation, and debugging of Windows applications. The program has been tested on a Pentium IV 3.06 Ghz (512 MB RAM). To obtain a feasible schedule, Hung's algorithm [11] calls for at least 30 minutes for a small size problem by hand whereas our program only requires a few seconds. The feasibility of these solutions can be confirmed by weekend and weekday requirement assignment and equality to the target requirements.

Illustrative example: $P = 3$ (shifts), $(A, B) = (1, 3)$ (at least one out of the three weekends off). Assume that the nurse requirements are D1=6, D2=5, D3=3, E1=5, E2=5, E3=3. The developed program computes the smallest workforce as $D = 14$, $E = 11$ and $Workforce = 24$.

We should use the following example to illustrate how steps of the algorithm are performed by the developed program. In the step 1, workforce size is computed and in the user interface, "0" represents off-days, "1" represents shift 1, "2" represents shift 2 and "3" represents shift 3. There are six requirement fields. Nurse requirements for each shift are obtained from the staff nurse or manager. In step 1, the smallest workforce size is computed by program. The smallest workforce size is calculated based on Hung's algorithm [11] that derived from literature [4, 5]. In Step 2, Off-weekends are assigned. In this step, $(W - E)$ nurses take weekend 1 off, the next $(W - E)$ nurses take weekend 2 off, and so on until $(W - E)$ nurses have been assigned to take weekend B off, where we wrap around (i.e. nurse 1 follows nurse W) when necessary. In step 3, *Off-Fridays* are assigned. Just after Step 2, in any week q, a nurse belongs to one of the four types; Type 1 is off on Sunday and off on Saturday. Type 2 is off on Sunday and at work on Saturday. Type 3 is at work on Sunday and off on Saturday. Type 4 is at work on Sunday and on Saturday. In each week, the program assigns Fri as an off-day until $(W - D)$ off-Fridays are given out, giving first priority to Type 4 nurses and second priority to Type 2 nurses. In step 4, additional off-weekdays are assigned. In each week, nurses take additional off-weekdays. If there are Type 2 nurses without an off-Friday, program assigns to these nurses Tue and Wed off. In step 5, Necessary weekend shifts and some weekday shifts are assigned. For the week 1. On Sun (Sunday) of week 1, program select E_j on-duty nurses who have not been assigned shifts on that day to shift j.

The program considers those Type 2 and Type 3 nurses concerned in week 1 in Step 4. Arbitrarily it associates each Type 2 with a Type 3 nurse. If a Type 3 nurse is on shift j on Sun, program gives nurse on shift j on Tue and Wed, and associated Type 2 nurse work on shift j on Mo, Th, Fr and weekend 1. For the week q ($q = 2,....B + 1$). The develop program sequentially works from $i = 2$. On weekend (i–1), it assigns just enough nurses who are on duty on the weekend to shift j to satisfy staffing requirements E_j. For $i \leq B$, the program considers those Type 2 and Type 3 nurses concerned in week q in Step 4. Program associates each Type 2 with a Type 3 nurse. If a Type 3 nurse is on shift j on Sun of week q, program gives work on shift j on Tue and Wed and associated Type 2 nurse work on shift j on Mon, Thu, Fri and weekend i. In step 6, Necessary weekday shifts are assigned. Some shift assignments have been already made on the ith day, $Sun \leq i \leq Thu$. There are nurses who have been assigned shifts on the ith day and who are also on duty on the (i+1)st day, but have not yet received shift assignments. If a nurse is on shift $j$on the ith day, program assigns the nurse to shift $j$on the (i+1)st day so that the nurse does not change shifts. In according to ergonomic rules, it is strongly desired that a nurse must be assigned same shifts consecutively. There is a small percent of consecutive days-on. It is

Compressed Workforce Program

Enter the count of workers at weekdays shift 1 : 6 Enter the count of workers at weekdays shift 2 : 5 Enter the count of workers at weekdays shift 3 : 3
Enter the count of workers at weekends shift 1 : 5 Enter the count of workers at weekends shift 2 : 5 Enter the count of workers at weekends shift 3 : 3

Prepare Clear Close

	Week 1							Week 2							Week 3						
	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	0	0	2	2	2	0	3	3	0	3	3	0	1	0	0	0	2	2	2	0	2
2	1	0	1	1	0	2	0	0	0	2	2	2	0	3	3	0	2	2	0	3	0
3	0	1	0	2	2	0	1	1	0	1	1	0	1	0	0	2	0	2	2	0	3
4	3	0	1	1	0	1	0	0	3	0	1	1	0	2	2	0	3	3	0	2	0
5	1	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	0
6	0	3	3	0	1	0	2	2	0	1	1	0	3	0	0	1	1	0	3	0	1
7	3	0	0	3	3	3	0	0	3	3	3	0	0	1	1	0	0	2	2	2	0
8	3	3	3	0	0	1	0	0	0	3	3	3	0	3	3	3	3	0	0	2	0
9	1	0	0	2	2	2	0	0	3	0	2	2	0	1	1	0	0	2	2	0	1
10	0	1	1	1	0	0	3	3	0	0	1	1	1	0	0	1	1	1	0	0	1
11	0	0	1	1	1	0	2	2	2	2	0	0	2	0	0	0	1	1	1	0	2
12	2	2	2	0	0	2	0	0	2	2	0	3	0	2	2	2	2	0	0	2	0
13	2	0	0	2	2	0	1	1	0	0	1	1	1	0	0	2	0	1	1	0	3
14	2	2	2	0	0	2	0	0	1	1	1	0	0	2	2	0	0	3	3	3	0
15	0	2	0	1	1	0	2	2	2	2	0	0	0	2	2	2	2	0	0	1	0
16	1	0	0	3	3	3	0	0	1	0	0	3	3	3	3	0	0	1	1	1	0
17	0	3	3	0	2	0	3	3	0	0	2	2	2	0	0	2	2	0	2	0	2
18	0	2	2	2	0	0	2	2	2	2	0	0	2	0	0	3	3	3	0	0	2
19	1	1	1	0	0	2	0	0	1	0	0	1	1	1	1	1	1	0	0	2	0
20	0	1	0	0	1	1	1	1	0	0	2	2	0	2	2	0	0	1	1	1	0
21	2	0	0	3	3	3	0	0	2	0	0	1	1	1	1	1	1	0	0	0	2
22	0	1	0	0	1	1	1	1	1	1	0	0	2	0	0	1	0	0	1	1	1
23	2	2	2	0	0	0	2	2	0	0	2	2	2	0	0	1	0	0	1	1	1
24	0	1	0	0	1	1	1	1	1	1	0	0	3	0	0	3	0	0	3	3	3

Fig. 1. A completed schedule for illustrative example

very normal assignment policy. This approach is get on well with ergonomic rule which support balancing among nurses.

Figure 1 displays a schedule generated by the developed program. There may be some nurses who have not been assigned weekday shifts. If a nurse is unassigned between two-off-days, program assigns nurse to shift j on the day(s) between. If a nurse is unassigned between an off-day and shift j, program assigns nurse to shift on the day(s) between. Program doesn't take into account nurses preferences for shifts or off-days on given days. If it does, it is nearly impossible to obtain an optimal schedule in this scheduling environment.

3 Conclusion

A new computer-assisted nurse scheduling program for multiple-shift workforce scheduling model under the compressed workweek has been just developed and presented. In the program, nurses are given 3 off days and 4 on days per week. The model assumes 3-workweek. The proposed program produces a schedule satisfying the daily demand while ensuring that the schedule is feasible, every nurse has at least Aout of B weekends off, each nurse takes exactly 4 days per from Sunday to Saturday, there are at least one day off between different shifts. The program was tested and our results were optimal like as Hung's algorithm.

References

1. Hung, R.: Multiple-shift workforce scheduling under the 3-4 workweek with different weekday and weekend labour requirements. Management Science 40(2), 280–284 (1994a)
2. Baker, K.: Scheduling a Full time Workforce to meet Cyclic Staffing Requirements. Management Science 20, 1561–1568 (1974)
3. Baker, K.R., Magazine, M.J.: Workforce scheduling with cyclic demands and day-off constraints. Management Science 24(2), 161–167 (1977)
4. Burns, R.N., Carter, M.W.: Workforce Size and Single Shift Schedules with Variable Demands. Management Science 31, 599–607 (1985)
5. Emmons, H.: Workforce scheduling with cyclic requirements and constraints on days off, and workstretch. IIE Transactions 17, 8–16 (1985)
6. Alfares, H.K.: Flexible 4-day workweek scheduling with weekend work frequency constraints. Computers and Indutrial Engineering 44, 325–338 (2003)
7. Alfares, H.K.: Dual-based optimization of cyclic four-day workweek scheduling. IMA Journal of Mathematics Applied in Business and Industry 11, 269–283 (2000)
8. Narasinhan, R.: Algorithm for multiple shift scheduling of hierarchical workforce on four-day or three-day workweeks. INFOR Journal 38(1), 14–32 (2000)
9. Burns, R.N., Narasimhan, R., Smith, L.D.: A set processing algorithm for scheduling staff on 4-day or 3-day work weeks. Naval Research Logistics 45, 839–853 (1998)
10. Lankford, W.M.: Changing schedules: A case for alternative work schedules. Career Development International 3, 161–163 (1998)
11. Hung, R.: A Multiple-shift workforce scheduling model under the 4- day workweek with weekday and weekend labour demands. Operational Research Society 45(9), 1088–1092 (1994b)
12. Hung, R., Emmons, H.: Multiple-shift workforce scheduling under the 3-4 compressed workweek with a hierarchical workforce. IIE Transactions 25(5), 82–89 (1993)
13. Nanda, R., Browne, J.: Introduction to employee scheduling. Van Nostrand Reinhold, New York (1992)
14. Arnold, B., Mills, M.E.: Core-12: Implementation of Flexible Scheduling. Journal Nurse Administration 13, 9–14 (1983)
15. Poor, R.: 4 days 40 hours. Mentor, Newyork (1973)
16. Steward, G.V., Larsen, J.M.: A four -day three-day per week application to a continuous production operations. Management Science 10, 13–20 (1971)
17. Hung, R.: Single-shift off-day scheduling of a hierarchical workforce with variable demands. European Journal of Operational Research 78, 49–57 (1994c)
18. Narasimhan, R.: Algorithm for a single shift scheduling of hierarchical workforce. European Journal of Operational Research 96, 113–121 (1997)
19. Billionnet, A.: Integer programming to schedule a hierarchical workforce with variable demands. European Journal of Operational Research 114, 105–114 (1999)

Postoptimal Analysis in Nonserial Dynamic Programming*

Oleg Shcherbina

Fakultät für Mathematik, University of Vienna
Nordbergstrasse 15, A-1090 Vienna, Austria
oleg.shcherbina@univie.ac.at
http://www.mat.univie.ac.at/~oleg

Abstract. Usually, discrete optimization problems (DOPs) from applications have a special structure, and the matrices of constraints for large-scale problems have a lot of zero elements (sparse matrices). One of the promising ways to exploit sparsity in the interaction graph of the DOP is nonserial dynamic programming (NSDP), which allows to compute a solution in stages such that each of them uses results from previous stages. The drawback of NSDP methods consists on exponential time and space complexity that is exponential in the induced width of the DOP's interaction graph. This causes an expediency and an urgency of development of tools that could help to cope with this difficulty. In this paper is shown that NSDP algorithm generates a family of related DOPs that differ from each other in their right-hand sides. For solving this family of related problems postoptimal and sensitivity analysis methods are proposed.

1 Introduction

Solving discrete optimization (DO) problems (DOPs) can be a rather hard task. Many real-life DOPs from applications contain a huge number of variables and/or constraints that make the models intractable for currently available solvers. Usually, DOPs from OR applications have a special structure, and the matrices of constraints for large-scale problems have a lot of zero elements (sparse matrices). One of the promising ways to exploit sparsity in the interaction graph of DOP is nonserial dynamic programming (NSDP), which allows to compute a solution in stages such that each of them uses results from previous stages.

In this paper is shown that NSDP algorithms generate a family of related DO problems that differ from each other in their right-hand sides. For solving this family of related problems postoptimal and sensitivity analysis methods are proposed.

* Research supported by FWF (Austrian Science Funds) under the project P17948-N13.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 308–317, 2008.

2 Discrete Optimization Problems and Their Graph Representations

Consider a DOP with constraints:

$$\max_X f(X) = \max_X \sum_{k \in K} f_k(X^k), \tag{1}$$

subject to

$$A_{iS_i} X_{S_i} \leq b_i,\ i \in M = \{1, 2, \ldots, m\}, \tag{2}$$

$$x_j = 0, 1,\ j \in N = \{1, \ldots, n\}, \tag{3}$$

where $X = \{x_1, \ldots, x_n\}$ is a set of discrete variables, functions $f_i(X^i)$ are called components of the objective function and can be defined in tabular form, $X^k \subset X,\ k \in K = \{1, 2, \ldots, t\}$, t is a number of components of the objective function,

$$S_i \subseteq \{1, 2, \ldots, n\},\ i \in M. \tag{4}$$

Definition 1. *[3]. Variables $x \in X$ and $y \in X$ **interact** in DOP with constraints if they appear both either in the same component of objective function, or in the same constraint (in other words, if variables both are either in a set X^k, or in a set X_{S_i}).*

Introduce a graph representation of a DOP. An **interaction graph** [3] represents a structure of the DOP in a natural way.

Definition 2. *[3]. The **interaction graph** of the DOP is called an undirected graph $G = (X, E)$, such that*

1. *Vertices X of G correspond to variables of the DOP;*
2. *Two vertices of G are adjacent iff corresponding variables interact.*

Further, we shall use the notion of vertices that correspond one-to-one to variables.

Definition 3. *The set of variables interacting with a variable $x \in X$, is denoted as $Nb(x)$ and called **neighborhood** of the variable x. For corresponding vertices a neighborhood of a vertex v is a set of vertices of interaction graph that are linked by edges with v. Denote the latter neighborhood as $Nb_G(v)$.*

Let S be vertex sets of the graph. Introduce the following notions:

Neighborhood of a set $S \subseteq V$, $Nb(S) = \bigcup_{v \in S} Nb(v) - S$;
Closed neighborhood of a set $S \subseteq V$, $Nb[S] = Nb(S) \cup S$.
If $S = \{j_1, \ldots, j_q\}$ then $X_S = \{x_{j_1}, \ldots, x_{j_q}\}$.

3 Graph Partitioning and Quotient Graphs

Let $G = (X, E)$ be an interaction graph of a DOP.

An **ordered partition** $\Pi = \{X_{K_1}, X_{K_2}, \ldots, X_{K_p}\}$ of a vertex set X is a decomposition of X into ordered sequence of pairwise disjoint nonempty subsets $X_{K_r},\ r = 1, \ldots, p$ whose union is all of X and $\cup_{r=1}^{p} K_r = N = \{1, \ldots, n\}$.

Finding algorithms that produce good partitions of graphs is more an art than a science. Several related decision problems are NP-complete (see, e.g., Arnborg et al. [2]) and hence unlikely to be solvable by fast algorithms. In practice one finds such partitions using greedy heuristics of minimum degree type (see, e.g., [1]) or of nested dissection type (see, e.g., **MeTiS** [7]).

Any partition induces an **equivalence relation**. Given a graph $G = (X, E)$, let Π be an ordered partition on the vertex set X:

$$\Pi = \{X_1, X_2, \ldots, X_p\}.$$

That is, $\cup_{i=1}^{p} X_i = X$ and $X_i \cap X_k = \emptyset$ for $i \neq k$. We define the **quotient graph** (George & Liu [5]) of G with respect to the partition Π to be the graph

$$G/\Pi = (\Pi, \mathcal{E}),$$

where $(X_i,\ X_k) \in \mathcal{E}$ if and only if $Nb_G(X_i) \cap X_k \neq \emptyset$.

Taking advantage of indistinguishable variables (two variables are **indistinguishable** if they have the same closed neighborhood (Amestoy et al. [1]) it is possible to compute a quotient graph, which is a more concise graph representation of the structure of the sparse problem. The quotient graph is formed by merging all vertices with the same closed neighborhoods into a single meta-node. Let X_k be a a block of a graph G (Arnborg, Corneil & Proskurowski [2]), i.e., a maximal set of indistinguishable with x vertices. Clearly, the blocks of G partition X since indistinguishability is an equivalence relation defined on the original vertices.

Consider below a NSDP block procedure [3].

4 Nonserial Dynamic Programming Block Elimination Scheme

One of the promising ways to exploit sparsity in the interaction graph of an optimization problem is NSDP (Bertele & Brioschi [3], Neumaier & Shcherbina [9]) which allows to compute a solution in stages such that each of them uses results from previous stages.

This approach is used in Artificial Intelligence under the names "variable elimination" or "bucket elimination". NSDP being a natural and general decomposition approach, considers a set of constraints and an objective function as recursively computable function. This allows one to compute a solution in stages such that each of them uses results from previous stages.

The efficiency of this algorithm crucially depends on the interaction graph structure of a DOP. The worst case performance of NSDP algorithms is exponential in the **induced width** of the interaction graph, also known as **tree-width**, namely the size of the largest cluster in an optimal tree-embedding of a graph.

Consider a DOP with constraints (1), (2), (3). The NSDP procedure can eliminate sets of variables ([3]).

Consider an ordered partition of the set X into blocks:

$$\Pi = (X_{K_1}, \ldots, X_{K_p}), \quad p \leq n.$$

For this ordered partition, the constrained DOP may be solved by NSDP.

A. Forward part

Consider first the block X_{K_1}. Then

$$\max_X \{C_N X_N | A_{iS_i} X_{S_i} \leq b_i,\ i \in M,\ x_j = 0,1,\ j \in N\} =$$

$$\max_{X_{K_2},\ldots,X_{K_p}} \{C_{N-K_1} X_{N-K_1} + h_1(Nb(X_{K_1})$$

$$| A_{iS_i} X_{S_i} \leq b_i,\ i \in M - U_1,\ x_j = 0,1,\ j \in N - K_1\}$$

where

$$U_1 = \{i : S_i \cap K_1 \neq \emptyset\} = U(K_1)$$

and

$$h_1(Nb(X_{K_1})) = h_1(X_{Nb(K_1)}) =$$

$$\max_{X_{K_1}} \{C_{K_1} X_{K_1} | A_{iS_i} X_{S_i} \leq b_i,\ i \in U_1,\ x_j = 0,1,\ j \in Nb[K_1]\}.$$

The first step of the block-elimination procedure consists in solving, using complete enumeration of X_{K_1}, the following optimization problem

$$h_1(Nb(X_{K_1})) = \max_{X_{K_1}} \{C_{K_1} X_{K_1} | A_{iS_i} X_{S_i} \leq b_i,\ i \in U_1,\ x_j = 0,1,\ j \in Nb[K_1]\}, \tag{5}$$

and storing the optimal partial solutions X_{K_1} as a function of a neighborhood X_{K_1}, i.e., $X^*_{K_1}(Nb(X_{K_1}))$.

The maximization of $f(X)$ over all feasible assignments $Nb(X_{K_1})$, is called the *elimination of the block* X_{K_1}. The optimization problem left after the elimination of X_{K_1}, is:

$$\max_{X - X_{K_1}} \{C_{N-K_1} X_{N-K_1} + h_1(Nb(X_{K_1}))\}$$

s.t.

$$A_{iS_i} X_{S_i} \leq b_i,\ i \in M - U_1,\ x_j = 0,1,\ j \in N - K_1.$$

Note that it has the same form as the original problem, and the tabular function $h_1(Nb(X_{K_1}))$ may be considered as a new component of the new objective function. Subsequently, the same procedure may be applied to the elimination of the blocks $X_{K_2}, \ldots, X_{K_p}$, in turn. At each step j the new component h_j and optimal partial solutions $X^*_{K_j}$ are stored as functions of $Nb(X_{K_j}$ |

$X_{K_1}, \ldots, X_{K_{j-1}}$), i.e., the set of variables interacting with at least one variable of X_{K_j} in the current problem, obtained from the original problem by the elimination of $X_{K_1}, \ldots, X_{K_{j-1}}$. Since the set $Nb(X_{K_p} \mid X_{K_1}, \ldots, X_{K_{p-1}})$ is empty, the elimination of X_{K_p} yields the optimal value of objective $f(X)$.

B. Backward part
This part of the procedure consists in the consecutive choice of $X^*_{K_p}, X^*_{K_{p-1}}, \ldots, X^*_{K_1}$, i.e., the optimal partial solutions from the stored tables $X^*_{K_1}(Nb(X_{K_1}))$, $X^*_{K_2}(Nb(X_{K_2} \mid X_{K_1})), \ldots, X^*_{K_p}$.

NSDP systematically proceeds with so called **parametric** DO problems [3]. An optimization problem is in p**arametric form** when the objective function is optimized not over entire set X, but only over a subset $X - P$, for all possible assignments of the variables of P. Below we show that this parametric form allows exploiting postoptimality and sensitivity tools in NSDP procedure.

5 Postoptimality Analysis

5.1 Postoptimality Analysis in DO

Decomposition and sensitivity analysis in DO are closely related. Sensitivity analysis follows naturally from the duality theory. Decomposition methods consist of generating and solving families of related DO problems that have the same structure but differ as the values of coefficients. Sensitivity analysis allows using information obtained during solving one DO problem of the family of related DO problems in solving other problems of this family. Due to the lack of full-fledged duality theory in DO, sensitivity analysis for DO problems is not sufficiently developed [4], [8]. A number of useful tools of sensitivity analysis in DO are derived for integer programming in [4]. A technique of sensitivity analysis proposed in [11] computes a piecewise linear value function that provides a lower bound on the optimal value that results from changing the right-hand sides of constraints. Recently, an interesting application of binary decision diagrams (BDD) (introduced earlier in computer science community) was proposed by Hadzic & Hooker [6] for the purposes of postoptimal analysis in DO. Particular implementation of postoptimality analysis depends on chosen computational DO algorithm (solver) and its properties.

5.2 Family of Related DO Subproblems in NSDP Block Procedure

Rewriting (5) we have the DOP

$$h_1(Nb(X_{K_1})) = \max_{X_{K_1}} \{C_{K_1} X_{K_1} | A_{iK_1} X_{K_1} \leq b_i - A_{iNb(K_1)} X_{iNb(K_1)},$$

$$i \in U_1,\ x_j = 0, 1,\ j \in K_1\}. \tag{6}$$

For all binary assignments of $Nb(X_{K_1})$ the parametric DOP (6) has to be solved. There is a family of related DO problems that differ from each other in their right-hand sides. For solving this family of related problems it is reasonable to use postoptimal and sensitivity analysis methods [4], [6], [8], [11].

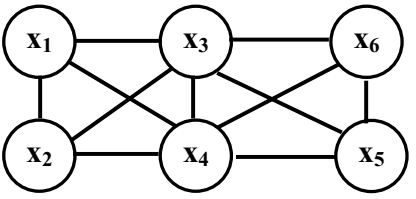


Fig. 1. Interaction graph in the DOP with constraints

Example 1. Consider a DOP (P) with binary variables:

$$\begin{aligned} 2x_1 + 3x_2 + \ x_3 + 4x_4 + 2x_5 + 4x_6 \quad &\to \max \ (OF) \\ x_1 + 2x_2 + 2x_3 + 3x_4 \quad &\le 6, \ (C_1) \\ x_3 + 2x_4 + 3x_5 + 2x_6 \quad &\le 6, \ (C_2) \\ x_j = 0, 1, \ j = 1, \ldots, 6. \end{aligned}$$

Apply the NSDP block procedure to the DO problem (P) from the example assuming $K_1 = \{1,\ 2\}$, $K_2 = \{5,\ 6\}$, $K_3 = \{3,\ 4\}$. Then $Nb(X_{K_1}) = \{x_3,\ x_4\} = X_{K_3}$.

The meta-DOP has the form:

$$\max_{X_{K_1}, X_{K_2}, X_{K_3}} \{C_{K_1} X_{K_1} + C_{K_2} X_{K_2} + C_{K_3} X_{K_3}\}$$

subject to

$$A_1^{(K_1)} X_{K_1} + A_1^{(K_3)} X_{K_3} \le 6,$$

$$A_2^{(K_3)} X_{K_3} + A_2^{(K_2)} X_{K_2} \le 6,$$

$$x_j = 0, 1, \ j = 1, \ldots, 6,$$

where

$$C_{K_1} = \begin{pmatrix} 2 & 3 \end{pmatrix}, \quad C_{K_2} = \begin{pmatrix} 2 & 4 \end{pmatrix}, \quad C_{K_3} = \begin{pmatrix} 1 & 4 \end{pmatrix},$$

$$A_1^{(K_1)} = \begin{pmatrix} 1 & 2 \end{pmatrix}, \quad A_1^{(K_3)} = \begin{pmatrix} 2 & 3 \end{pmatrix}, \quad A_2^{(K_3)} = \begin{pmatrix} 1 & 2 \end{pmatrix}, \quad A_2^{(K_2)} = \begin{pmatrix} 3 & 2 \end{pmatrix},$$

$$X_{K_1} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad X_{K_2} = \begin{pmatrix} x_5 \\ x_6 \end{pmatrix}, \quad X_{K_3} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}.$$

Let us formulate a family of related DO subproblems corresponding to a block X_{K_1}:

$$h_1(Nb(X_{K_1})) = h_1(x_3,\ x_4) = \max_{X_{K_1}} \left\{ C_{K_1} X_{K_1} | A_1^{(K_1)} X_{K_1} \le b - A_1^{(Nb(K_1))} X_{Nb(K_1)} \right\} =$$

$$\max_{x_1,\ x_2} \{2x_1 + 3x_2 | x_1 + 2x_2 \le 6 - 2x_3 - 3x_4\}. \tag{7}$$

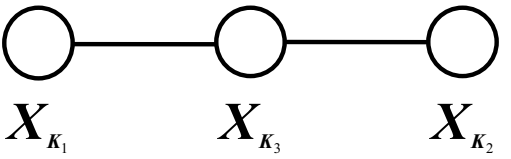


Fig. 2. Quotient graph

This is a family of knapsack problems. To solve these problems let us solve only one DOP of the family, namely the knapsack problem with maximum right-hand sides (if $x_3 = x_4 = 0$) using usual dynamic programming procedure:

$$h_1(0,0) = \max_{x_1,\ x_2} \{2x_1 + 3x_2 | x_1 + 2x_2 \le 6,\ x_1, x_2 = 0, 1\}.$$

Bellman's recursive equation for solving the knapsack problem

$$\max\left\{\sum_{j=1}^{n} c_j x_j \mid \sum_{j=1}^{n} a_j x_j \le b,\ x_j = 0, 1,\ j = 1, \ldots, n\right\}$$

(where $c_j,\ a_j,\ b$ are integer) is:

$$f(k, y) = \max\{f(k-1, y),\ f(k-1, y-a_k) + c_k\},$$

where

$$f(k, y) = \max\left\{\sum_{j=1}^{k} c_j x_j \mid \sum_{j=1}^{k} a_j x_j \le y,\ x_j = 0, 1,\ j = 1, \ldots, k\right\},$$

$$f(0, y) = 0,\ f(k, 0) = 0.$$

All objective function values and solutions for $h_1(x_3, x_4)$ are contained in this table.

Table 1. Calculation of $h_1(0,0)$

$k\backslash y$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	2*	2*	2*	2*	2*	2*
2	0	2	3*	5*	5*	5*	5*

Remark1. Value with "star" in the tables (e.g., 5^*) means that corresponding variable x equals 1.

Thus, we got a tabular function $h_1(x_3, x_4)$:

Table 2. Calculation of $h_1(x_3, x_4)$

x_3	x_4	h_1	$x_1^*(x_3, x_4)$	$x_2^*(x_3, x_4)$
0	0	5	1	1
0	1	5	1	1
1	0	5	1	1
1	1	2	1	0

Consider the next block X_{K_2}. Let us formulate a family of related DO subproblems corresponding to a block X_{K_2}:

$$h_2(Nb(X_{K_2})) = h_2(x_3,\ x_4) = \max_{X_{K_2}} \left\{ C_{K_2} X_{K_2} | A_1^{(K_2)} X_{K_2} \le b_2 - A_1^{(Nb(K_2))} X_{Nb(K_2)} \right\} =$$

$$\max_{x_5,\ x_6} \{2x_5 + 4x_6 | 3x_5 + 2x_6 \le 6 - x_3 - 2x_4\}. \tag{8}$$

This is a family of knapsack problems. To solve these problems let us solve only one DOP of the family, namely the knapsack problem with maximum right-hand sides (when $x_3 = x_4 = 0$) using usual dynamic programming procedure:

$$h_2(0,0) = \max_{x_5,\ x_6} \{2x_5 + 4x_6 \mid 3x_5 + 2x_6 \le 6,\ x_5, x_6 = 0, 1\}.$$

All objective function values and solutions for $h_2(x_3, x_4)$ are contained in this table.

Table 3. Calculation of $h_2(0, 0)$

$k \backslash y$	1	2	3	4	5	6
1	0	0	2*	2*	2*	2*
2	0	4*	4*	5*	6*	6*

Thus, we got a tabular function $h_2(x_3, x_4)$:

Table 4. Calculation of $h_2(x_3, x_4)$

x_3	x_4	h_2	$x_5^*(x_3, x_4)$	$x_6^*(x_3, x_4)$
0	0	6	1	1
0	1	4	0	1
1	0	6	1	1
1	1	4	0	1

Consider the last block X_{K_3}. We have the DOP:

$$\max_{x_3, x_4} \left[h_1(x_3, x_4) + h_2(x_3, x_4) + x_3 + 4x_4 \right].$$

Optimal solution is: $x_3^* = 0,\ x_4^* = 1$ with objective function value 13.

After backward step of the NSDP we have: Table 4: $x_5^* = 0,\ x_6^* = 1$. Table 2: $x_1^* = 1,\ x_2^* = 1$.

The solution is (1, 1, 0, 1, 0, 1), the maximum objective value is 13.

Postoptimal analysis was here practically trivial, because we used the properties of dynamic programming tables for knapsack problems and calculate only one table for each block $X_{K_1},\ X_{K_2}$.

In more general case it is possible to introduce a partial order over a family of related DO problems. This allows to solve the members of the family in a corresponding sequence to yield useful information and to take advantage of the information generated by the solution to one member of the family in order to reduce the running time necessary to solve another member.

NSDP algorithms combined with modern DO solvers are a promising approach that enables solving sparse discrete optimization problems from applications. The performance of these algorithms can be improved with the aid of postoptimality analysis.

Promising direction of future research is the development of efficient schemes of postoptimality analysis embedded in NSDP algorithms combined with DO solvers.

References

1. Amestoy, P.R., Davis, T.A., Duff, I.S.: An Approximate Minimum Degree Ordering Algorithm. SIAM J. on Matrix Analysis and Appl. 17, 886–905 (1996)
2. Arnborg, S., Corneil, D.G., Proskurowski, A.: Complexity of Finding Embeddings in a k-Tree. SIAM J. Alg. Discr. Meth. 8, 277–284 (1987)
3. Bertele, U., Brioschi, F.: Nonserial Dynamic Programming. Academic Press, New York (1972)
4. Geoffrion, A.M., Nauss, R.: Parametric and Postoptimality Analysis in Integer Linear Programming. Management Science 23, 453–466 (1977)
5. George, A., Liu, J.W.H.: A Quotient Graph Model for Symmetric Factorization. In: Duff, I.S., Stewart, G.W. (eds.) Sparse Matrix Proceedings, pp. 154–175. SIAM Publications, Philadelphia (1978)
6. Hadzic, T., Hooker, J.: Cost-bounded Binary Decision Diagrams for Programming. In: Van Hentenryck, P., Wolsey, L.A. (eds.) CPAIOR 2007. LNCS, vol. 4510. Springer, Heidelberg (2007)
7. Karypis, G., Kumar, V.: MeTiS – a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Version 4, University of Minnesota (1998), http://www-users.cs.umn.edu/~karypis/metis
8. Marsten, R.E., Morin, T.L.: Parametric Integer Programming: The Right Hand Side Case. In: Hammer, P., et al. (eds.) Annals of Discrete Mathematics: Studies in Integer Programming. Elsevier, North Holland (1977)
9. Neumaier, A., Shcherbina, O.: Nonserial Dynamic Programming and Local Decomposition Algorithms in Discrete Programming. Discrete Optimization (submitted), http://www.optimization-online.org/DB_HTML/2006/03/1351.html

10. Parter, S.: The Use of Linear Graphs in Gauss Elimination. SIAM Review 3, 119–130 (1961)
11. Schrage, L.E., Wolsey, L.A.: Sensitivity Analysis for Branch and Bound Integer Programming. Operations Research 33, 1008–1023 (1985)
12. Shcherbina, O.: Nonserial Dynamic Programming and Tree Decomposition in Discrete Optimization. In: Proceedings of Int. Conference on Operations Research "Operations Research 2006", Karlsruhe, 6–8 September, 2006, pp. 155–160. Springer, Berlin (2007)

A Novel Optimization in Guillotine Cut Applied Reel of Steel

Plácido Rogério Pinheiro, José Aelio Silveira Junior, João Batista Furlan, Clécio Tomaz, and Ricardo Luiz Costa Hollanda Filho

Universidade de Fortaleza, Mestrado em Informática Aplicada
Av. Washington Soares 1321, Sala J-30, Fortaleza, CE, Brasil, 60811-905
{placido,furlan}@unifor.br, aelio@esmaltec.com.br, clecio@larces.uece.br, rick1700@gmail.com
http:www.unifor.br/mia

Abstract. A new framework for a problem of guillotined cut and selection in stock of reels of steel of the industry metal-mechanics formulated is present. Initially a model of lineal programming aiming at to minimize the losses of the lengths of the cut plans, afterwards a model of integer lineal programming completes of selection of reels applied to the stock that assists the demand, minimizing the production surpluses and the time with the logistics interns (transport and movement of reels).

1 Introduction

The new environment of competitiveness, caused for the growth, globalization and evolution of the economy, imposes now, that the industries have a still bigger commitment with it continue perfectioning of its products, processes and elimination of wastefulness waste during the productive stage. There is one considered publication number exploring the possibilities, not only the theoretical problems, but, overall industrial applications [3]. In this context a modeling in integer linear programming for optimization of cut in steel reels is considered, whose objective is to minimize the losses in raw material, the excesses of production and the time with the logistic intern (transport and movement of reels).

2 The Problem of Guillotine Cut and Selection of Reels in Stock

In the planning of the production of some industrial segments, the objective is to minimize the negative effect generated by wastefulness of materials and equally excellent in the planning of logistic operations as storage, movement and transport, aiming at to the movement of idle spaces [6]. For [7], beyond the loss of material when the objects biggest are cut in itens lesser, will also be admitted a cost associated with the preparation of the production of each product in determined periods of planning. With this an economic pressure appears to manufacture some end items anticipated in order to minimize the loss

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 318–327, 2008.

and the costs of preparation, respecting the available capacity in each period of the planning horizon. To justify the necessity of the stock of reels some factors must be considered: Slow transport between siderurgy and the plant due to the difficulties in the transport of the reels. The storage places must be great the sufficient to store the reels. However the logistic intern (transport between stock until the cut machines) must be considered, what she becomes inevitable the piling up of the reels. After the election of the reels in stock, the operator of the transporters must carry through the movements of displacement of the stock of the reels for the cut machine (to slitter). The times, selected reel met in position of difficult access (some reels piled up on the selected reel. Such fact makes it difficult the operation of withdrawal of the selected reels, therefore a bigger number of movements to be is necessary carried through for the transporter.

3 Cutting of Steel Reels

A reel is uncurled and the process of cut for attainment of the intermediate reels is made longitudinally by "cut records" (it does not have transversal cuts and therefore it can be understood that the problem is one dimension). Intrinsic losses [5] in the laterals of the reels exist, to eliminate the irregularities of the edges, varying enter 3 to 6 mm for edge. The time of preparation of the cut machine (Slitter) is about 50 minutes and the average time for cut of a funny reel around 20 minutes. After the cut, the reels by the proper machine to slitter, giving origin to the straps (or reels you would intermediate) that they will be referenced as reels blanks. The reel blank will be able to follow two ways: stored in the stock of the reels blanks (cut reels already) or to be directed the production where they will suffer as a cut in the machine, giving origin blanks . For reasons techniques the possible number of straps to be cut by reel is inversely proportional to the gauge. To the widths differentiated of blanks (parts), the reels can be cut in different ways. In the practical one, by a restriction in the total number of knives (cut records) and functioning of the Slitter, the amount of reels blanks generated by cut standard was limited (maximum of 30 reels blanks).

4 The Model of Guillotine Cut and Selection of Reels in Stock

To determine a standard of cut for unidimensionais problems definition (1) consists, in inside placing lesser units (item) of a bigger unit (object) of the possible form most valuable. From a reel B, of width ℓ continuous, is desired to cut I reels blanks b_{li} of width $\ell_i \geq \ell$, and length L_i, $i = 1, ..., I$. Each reel blank b_{li}, it will suffer after that to one guillotine cut to give length parts. The value is determined of L, total length of B, thus as the used standards of cut. These standards of cut are only the different possibilities of disposal of n reels blanks b_{li}, in the width of the main reel B. To each standard of cut s, we associate $n_{i(s)}, i = 1, ..., I$, the number of reels blanks type b_{li} contained in s.

Definition 1: A standard of cut s, is said possible if the addition of the widths of the reels blanks b_{li}, that it composes it, the width is inferior ℓ of the reel B, which is:

$$p(s) = \ell - \sum_{i=1}^{I} n_{i(s)} * \ell_i \tag{1}$$

One denotes p(s), the corresponding loss to this standard of cut, whose value of p(s) $\geq$0.

Definition 2: A cut standard s it is viable if it is possible and p(s) is inferior the lesser width of the reels blanks b_{li}:

$$0 \leq p(s) \leq \min_{i=1,\ldots,I} \ell_i \tag{2}$$

The number of possible combinations, the process of generation of columns can be become a critical operation. To optimize the processing time [1] was decided to use an algorithm in tree. The viable considered standards of cut will be considered by the cut model. Some parts exist that only can to be cut in one of the directions (length or width), had as cut in the machine to be of the type when all the process of cut in the machine uncurls the reel blank to the cut that will give origin blank, it is carried through of automatic form To treat these particularitities, the cut standards are generated considering all the possible combinations, being in charge of the model to choose the direction to be cut of the parts that do not possess restriction how much to the cut in the machine. Either x_{bj} the excellent length of the standards of cut necessary to take care of demand L_i. The cut of the steel reels can be formulated:

$$\min \sum_{b\in B} \sum_{j\in J} \left(p_{bj} + \sum_{i\in I} \frac{n_{ij}}{pas_i} \right) * x_{bj} \tag{3}$$

subject a:

$$\sum_{b\in B} \sum_{j\in J} \frac{n_{ij}}{pas_i} * x_{bj} \geq L_i, i = 1, ..., I, x_{bj} \geq 0, j = 1, ..., J. \tag{4}$$

Where: J - number of viable standards of cut; B - number of types of reels; I - types of parts necessary to take care of the demand; x_{bj} - length of the cut standard s_j, $j = 1, ..., J$ in reel b; p_{bj} - loss p (s_j) associated to the standard s_j, $j = 1, ..., J$ in reel b; $n_{ij} = n_i(s_j)$ - amount of straps of part i of standard j; pas_i, corresponds the width or the length of the part, depending on the choice of standard. The objective function (3) to minimize the total loss (p_j), and the excess of parts gotten, represented for $(\sum_{i\in I} \frac{n_{ij}}{pas_i})$. The restrictions (4) correspond to the minimum lengths, that if must cut for each type of part, of form that if takes care of the demand. The great volume of reels in stock, it is necessary to formulate a model being objectified to support the sector of planning and control of the production in the choice of the reels that must be cut. After to have joined the length of the excellent standards through model P_{cut}, that to

inform which reels will have to be selected in form stock that is possible to cut them with the exceeding minor of reels blanks.

$$\min \sum_{j \in J} \sum_{k \in K} (T_k \; y_{jk} - x_{jk}) \tag{5}$$

subjetc to:

$$\sum_{k \in K} x_{jk} \geq C_j, j = 1, ..., J \tag{6}$$

$$\sum_{j \in J} x_{jk} \leq T_k, k = 1, ..., K \tag{7}$$

$$\sum_{j \in J} y_{jk} \leq 1, k = 1, ..., K \tag{8}$$

$$x_{jk} - fator\_minT_k y_{jk} \geq 0, T_k y_{jk} \geq 0, j = 1..J, k = 1, ..., K \tag{9}$$

$$x_{jk} \geq 0, y_{jk} \in \{0, 1\}, j = 1, ..., J, k = 1, ..., K$$

Where: T_k - length of the reel b_k available in stock, C_j - length of the cut standard s_j, $j = 1, ..., J$ found in model P_{cut}, x_{jk} - length of standard j to be cut effectively of the reel b_k; y_{jk} it is equal 1 (one) if reel b_k it is cut for standard j, and equal the 0 (zero) in contrary case, fator_min it is a parameter that informs the minimum percentage to be cut of reel bk for standard j. Value configured and only for all reels b_{jk}. The objective function (5) aims at to minimize the lengths of the reels in stock ($T_k \; y_{jk}$), while it maximizes the lengths of the cut standards ($-x_{jk}$) found in model (P_{cut}). The intended to diminish the excesses of reels blanks had not the use of the reel in stock for complete. The restrictions (6) they assure that the total length to be cut of reel b_k, for a cut standard j, it takes care of to the length gotten for model (P_{cut}). The restrictions (7) impose that length x_{jk} effectively to be cut of reel bk either lesser that the total length of the reel in stock (T_k). In (8) one inquires that the reel will be cut by an only standard of cut. The minimum length (given a minimum factor) and that the maximum length, to be cut of reel b_k, they do not exceed the lengths of reels b_k in stock (T_k) it is assured in the restrictions (9).

5 Application in the Metal-Mechanics Industry

The Esmaltec S/A located in Maracanau/Ceara/Brazil, an industry metal-mechanics with production of stoves, coolants, to freezer, water through and containers GLP being aimed at to take care of the domestic market and external. It needs to carry through cut to produce a plan of monthly production, as Table 1. It considers the parts of the type of plate 0.75 mm, as Table 1, one has storage with 35 reels of 0.75 mm x 1.200 mm and lengths varied given in Table 3.

Table 1. Plan of monthly production

Production Plan	Quantity	Production Plan	Quantity
Support Baby Ch-3 White	5000	Horizontal Freezer 2255ect1	10000
Support Baby Ch-3 Inox	1000	Vertical Freezer 3050ec	2000
Stove Pantanal 0294 White	1000	Horizontal Freezer 3168ect1	200
Stove Taiba 4207	25000	Horizontal Freezer 3169ect2	600
Stove Itapuã 4311	30000	Horizontal Freezer 3165ect2	1000
Stove Olinda 4407	5000	Horizontal Freezer 4505ect2	1000
Stove Angra 4607	3184	Gelágua Gnc - 7be	5000
Stove Maresias 470	1246	Gelágua Gnm-1be	5220
Stove Salina 4803	2000	Gelágua Gnc - 1ae	2300
Stove Olinda 6405	2000	Gelágua Gn - 97ce	500
Stove Angra 6605	5862	Gelágua Gnc - 1ae Inox	1000
Stove Maresias 6705	2000	Refrigerator - RUP2450ec	3000
Stove Salinas 6803	4328	Refrigerator - RUP3100ec	2000
Refrigerator - RDP3140ec	3000		

Table 2. Description of parts 0.75 mm to be produced

Description of the piece	Type of steel	Thickness (mm)	Width (mm)	Length (mm)	Pieces for blank	Number of pieces	Demand
Comp. Profile Drain plug 3mm	1	0.75	120	33	1	2	132.86
Comp. Profile Drain plug 4mm	1	0.75	120	33	1	3	102.57
Comp. Knob 6q	1	0.75	770	98	1	1	20.00
Normal Espalhador	1	0.75	84	83	1	2	2.00
Door Fv340	1	0.75	661	1573	1	1	2.00
Knob Oven 6	1	0.75	810	106	1	1	20.00
Reinforcement Door Fv340	1	0.75	84	1440	1	2	4.00
Normal ice-cream dealer	1	0.75	75	72	1	2	2.00
Sup/Inf - Burning Oven	1	0.75	350	240	2	2	132.86
Come in sight drain plug 2255	1	0.75	717	853	1	1	10.00
Come in sight drain plug 3165	1	0.75	717	581	1	2	2.00
Come in sight drain plug 3167	1	0.75	717	1086	1	1	200.00
Come in sight drain plug 4505	1	0.75	717	756	1	2	2.00

It considers the parts Table 2 to be produced to take care of to the plan of production Table 3 and the listing of the reels B (0.75 x 1.200) available in stock Table 5.

Table 3. Stock of reels B

Reel	Weight	Length	Reel	Weight	Length	Reel	Weight	Length	Reel	Weight	Length
1	12.650	1.757	10	11.450	1.590	19	10.000	1.389	28	10.000	1.389
2	13.170	1.829	11	12.310	1.710	20	10.000	1.389	29	10.000	1.389
3	11.390	1.582	12	11.190	1.554	21	10.000	1.389	30	10.000	1.389
4	12.230	1.699	13	13.590	1.888	22	10.000	1.389	31	10.000	1.389
5	7.820	1.086	14	12.820	1.781	23	10.000	1.389	32	10.000	1.389
6	10.760	1.494	15	10.000	1.389	24	10.000	1.389	33	10.000	1.389
7	13.730	1.907	16	10.000	1.389	25	10.000	1.389	34	10.000	1.389
8	12.750	1.771	17	10.000	1.389	26	10.000	1.389	35	10.000	1.389
9	12.720	1.767	18	10.000	1.389	27	10.000	1.389			

The table 4 presents the results of sequential models (P_{cut} e P_{sel}). The computational tests had been carried through in a computer Pentium IV 3.0 Ghz, with 1 Mb of memory, applying Java, with CPLEX.

Table 4. Result of sequential model (P_{cut} e P_{sel})

	Simulation	Model	Iteration	Objective	Status	Time(sec)	Lines	Columns
1	Production Plan(Table3)	P_{cut}	41	374,502	Optimo	5	13	1006
		$P_{sel}(11standard)$	415	0	Optimo	3	852	770
					Total	8		

On the other hand, Table 5 presents the consumption of the steel (kg), the loss (kg) and the scrap iron index (%) generated the execution of models P_{cut} and P_{sel}.

Table 5. Demonstrative of consumption, loss and scrap iron of the steel plate

Reel	Weight(mm)	Length(mm)	Loss(kg)	Consumption(kg)	Waste
5,010,090,364	0.75	1,200	7,845.00	160,002.05	4.90%

Considering the losses associates to each standard of cut, a loss is registered of 7.845 Kg of a total of 160.000 Kg of steel consumption. They have a scrap iron of 4,9% of the steel consumption considering the plan of production of Table 5. An initiated time the cut, the machine (to slitter) it can stop at any time, but the operation that consists of modifying the blades (cut knives) of position is delayed. To consider a model in which it is considered that the machine alone will be able to stop when all the reel will be cut. The inconvenience biggest of this attempt is that in practical it generates new excesses of reels blanks.

6 Model for Reels Blanks in Stock

In the case of the industry, it is looked exploitation of the reels blank in stock, objectifying a reduction of the losses and a reduction of the production excesses.

All the reels (B) selected for model P_{sel} they are cut completely had to the raised involved cost in the operation of exchange of "records of cut". They have a number of reels blanks (b_i) with bigger length that the demand. The excess does not have to suffer as the cut in the machine, therefore blanks that they will not be used in the production immediately they still have a bigger probability of compared oxidation if with the storage in reels blanks. Such fact generates an excess of reels blanks (b_i) with characteristics of width, definite thickness and lengths. Where situations must be used the reels blanks in stock without generating new excesses of production? One presents a model for the problem, P_{stock} , with intention to use to advantage the reels blanks stored in stock. The application of the model of exploitation of stock (P_{stock}) is conditional the situations:

When the length of the reels blanks in stock it is minor of what the length of blanks of the demand;

When the length of the reels blanks in stock it is greater that the length of blanks of the demand. In the first case the reels blanks the exploitation and the balance of the amounts will be chosen for of blanks it will go to compose the demand for model P_{cut}.The reels blanks in stock already the width or length of the parts is cut in straps in agreement. Initially the width of the part is carried through in agreement search to be produced. In case that no reel does not exist blank in stock that satisfies the condition for the width, a new search is carried through having considered the length of the part to be produced. After that, a model is presented to optimize the excess of blanks generated as the choice of the reel blank in stock, a time that any reel blank chosen in stock it possess the length biggest that the total demand of blanks. The part that will be taken care of completely will not go to compose the restriction in the model of cut (P_{cut}).

$$\min \sum_{i \in I} q_i y_i \tag{10}$$

$$subject\,to:$$

$$\sum_{i \in I} q_i y_i \geq L_i \tag{11}$$

$$y_i \in \{0,1\}, i = 1, ..., I$$

Where: q_i amount of blanks in stock that could be produced by the reel blank b_{li}; L_i demand of blanks to be produced by the reel blank b_i ; y_i equal the 1 if reel blank b_{li}; it will be chosen, and equal the 0 in contrary case. The objective function aims at to minimize the amount of blanks (ud) to be produced. The restrictions they assure that the amount of blanks to be used it takes care of the total demand without generating new excesses of blanks e the restrictions guarantee which reel must be chosen.

7 Computational Results

It considers the scene in which if they have in stock of 12 reels blanks as Table 6.

Table 6. Relation of the reels blanks in stock

Blank Reel	Thickness(mm)	Width(mm)	Type of steel	Weight(Kg)	Length
1	0.75	120	1	1.200	1.666,67
2	0.75	120	1	1.200	1.666,67
3	0.75	661	1	1.000	252,14
4	0.75	661	1	1.000	252,14
5	0.75	84	1	900	1.785,71
6	0.75	84	1	900	1.785,71
7	0.75	717	1	2.000	464,9
8	0.75	717	1	2.000	464,9
9	0.75	350	1	1.500	714,29
10	0.75	350	1	1.500	714,29
11	0.75	810	1	1.200	246,91
12	0.75	810	1	1.200	246,91

The table 7 shows the results of sequential models (P_{cut} e P_{sel}) considering the exploitation of the reels blanks in stock.Comparing the data of first simulation Table 4 with the results of second simulation Table 7, observes a profit in the objective function, time of processing and number of iterations of models P_{cut} e P_{sel} . The table 8 presents the consumption of the steel (kg), the loss (kg) and the scrap iron index (%) generated execution of models P_{stock}, P_{cut} after P_{sel}.

Table 7. Exploitation of reels blanks in stock

	Simulation	Model	Iteration	Objective	Status	Time(sec)	Lines	Columns
2	Production Plan(Table3)	P_{cut}	26	258,804	Optimo	3	13	266
		$P_{sel}(10standard)$	388	0.9094	Optimo	2	781	770
					Total	5		

Table 8. Demonstrative of consumption, loss and scrap iron of the steel plate

Reel	Weight(mm)	Length(mm)	Loss(kg)	Consumption(kg)	Waste
5,010,090,364	0.75	1,200	5,921.64	137,398.70	4.31%

The loss using the exploitation of the reels blanks (model P_{stock}) it was 5.921,64 Kg, that guarantees a real profit of 1.924 Kg or 24,52% if compared with the model without exploitation of reels blanks. Considering the consumption of 137.398 Kg, registers a profit of 22.604 Kg, or 14,13%. Consequently they get a reduction of 0,6% in the generation of the sucatas considering only the reels of the type of plate 0.75 mm.

7.1 Comparative Analyses of the Results

A sequence of comparative tests between models P_{cut} is presented cut (without stock exploitation) and the P_{cut} using considered model P_{stock} (with stock

Table 9. Comparative simulations between models C_{est} e S_{est}

			With Stock - Cest			Without Stock -Sest		
	Plan of Production	Stock Reels	Losses (Kg)	Consumption (Kg)	Waste	Losses (Kg)	Consumption (Kg)	Waste
1	144.400	35	5.922	137.399	4,31%	7.845	160.002	4,90%
2	146.300	40	5.503	140.200	3,93%	6.840	162.120	4,22%
3	149.150	45	6.200	143.594	4,32%	7.200	165.980	4,34%
4	151.650	50	5.130	155.035	3,31%	6.780	168.340	4,03%
5	154.040	55	6.501	161.040	4.04%	7.650	170.870	4,48%
6	157.980	60	6.300	163.050	3,86%	6.700	172.345	3,89%
7	160.200	65	6.900	168.340	4,10%	7.990	173.250	4,61%
8	163.480	70	5.500	170.200	3,23%	6.990	175.238	3,99%
9	166.780	75	7.100	173.105	4,10%	7.890	177.654	4,44%
10	169.345	80	5.830	175.055	3,33%	7.010	179.486	3,91%
11	172.045	85	7.250	171.230	4,23%	7.930	180.521	4,39%
12	175.200	90	6.990	170.300	4,10%	7.500	181.450	4,13%
13	177.400	95	7.000	175.340	3,99%	7.350	182.340	4,03%
		Total	82.126	2.103.888	3,90%	95.675	2.249.596	4,25%

exploitation). It considers C_{est} the model with exploitation of stock e S_{est} the model without stock exploitation . In the Table 9 data of the 13 carried through simulations modifying the plan of production and the amount of reels in stock.

A profit is had when we apply the model C_{est}. Beyond the profit in the cited previously, one better control of the reels is obtained blanks in stock and consequently one better planning of the production.

8 Conclusion

The carried through computational tests with data of Esmaltec S/A, had presented the evolution of the models P_{cut} and P_{sel} to the being used with the P_{stock} model. Analyzing the results, it is verified that the application of the P_{stock} model reflects in the reduction of the pointers of loss, scrap iron and consumption. The simulations carried through with the parts of the type of plate 0.75mm had presented an average reduction of 13.500 kg (14%) in losses and 145.000 kg (6%) in the consumption of plates. It is standed out reduction with other types of plate that are used in the products of the Esmaltec, for example: the plates with thickness of 0.50mm, 0.55mm, 0.60mm, 0.65mm and 1.50mm. Important to cite that beyond the economy in the consumption of the plate of steel, indirect costs as: loss for oxidation of reels blanks in stock, logistic intern (transport of the reels), energy and man power had been also reduced with the job of this model. For future works it is considered to take care of the plan of sales in one determined period, analyzing the variable of time of preparation of machine, cost of stock, wallet of order and loss of the cut standards.

References

1. Chvatal, V.: Linear Programming. Mc Gill University, EUA (1983)
2. Golden, B.: Approaches to cutting stock problem. AIIE Transaction. 8, 265-274 (1976)
3. Nepomuceno, N. V., Pinheiro, P. R., Coelho, A. L. V.: Tackling the Container Loading Problem: A Hybrid Approach Based on Integer Linear Programming and Genetic Algorithms. In: Cotta, C., Hemert, J. (eds.) EvoCOP 2007. LNCS, vol. 4446, pp. 154-165. Springer, Heidelberg (2007)
4. Nepomuceno, N. V., Pinheiro, P. R., Coelho, A. L. V.: A Hybrid Optimization Framework for Cutting and Packing Problems: Case Study on Constrained 2D Non-guillotine Cutting. In: Cotta, C., Hemert. J., (eds.) Recent Advances in Evolutionary Computation for Combinatorial Optimization.: Springer, Heidelberg to appear (2008)
5. Wascher, G., Gau, T.: Heuristics for the integer one-dimensional cutting stock problem: A computational study, OR Spectrum. 18(3), 131-144 (2005)
6. Karelahti, J.: Solving the Cutting Stock Problem in the Steel Industry. Helsinki University, Master's Thesis (2002)
7. Haessle, R.W., Vonderembse, M. A.,: A Procedure for Solving the Master Slab Cutting Stock Problem in the Steel Industry. IIE Transactions, 11(2), 160-165 (1979)

Robust Production Planning: An Alternative to Scenario-Based Optimization Models

Carles Sitompul and El-Houssaine Aghezzaf

University of Ghent, Department of Industrial Management
Technologiepark 903, B9052 Zwijnaarde, Belgium
{Carles.Sitompul,ElHoussaine.Aghezzaf}@ugent.be

Abstract. Robust planning approaches, which specifically address the issue of uncertainty in production systems, are becoming more and more popular among managers. Production planning models which incorporate some of the system's uncertainty, at earlier stages in the planning process, are capable of generating 'stable' plans that are robust to the variability resulting from some critical planning parameter. In this paper we review some models for robust planning and their solution approaches. We then propose and discuss a new alternative model for aggregate production planning when periodic demands are uncertain. The objective of the model is to provide cost effective production plans while maintaining the targeted service levels. The performance of the proposed alternative model is compared with that of the scenario-based optimization models, and the obtained results are thoroughly discussed.

Keywords: Robust planning, scenario optimization.

1 Introduction

Uncertainty is present at all levels in a production system. This uncertainty may result from machine breakdowns, processing capabilities or human failures. If it is not taken into account during the planning, the system's performance may extremely deteriorate. The service level may dramatically decline when the actual demand is higher or when the supplier's lead time is longer than expected. In order to protect the system against these uncertainties, a course of actions is needed. Many efforts and research have been made to address this issue, in particular using a technique called buffering. This technique includes safety stock, safety lead-time and safety capacity (see Guide & Srivastaya, [3] for an extensive survey). Until recently, sensitivity analysis is used as post-optimality studies to discover the impact of data variability on the model's recommendation. However, this does not solve the issue since it is a rather passive approach. We actually need a proactive approach which can produce solutions that are less sensitive to the data variability. Robust model incorporating the uncertainty into the model itself seem to carry the answer to this issue.

Mulvey *et al.* ([8]) defined a robust optimal solution as one that remains 'close' to optimal for all scenarios of the input data, and model-robust solution

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 328–337, 2008.

as one that remains 'almost' feasible for all data scenario. The optimal value of the decision variables (design variables) is not conditioned on the realization of the uncertain parameters, the control variables, however are subjected to adjustments once the uncertain parameter is observed. Using multiple scenarios, the objective becomes a random variable where the mean value is used as the objective function in the stochastic linear programming. The approach of robust planning also handles risks or higher moments of the objective function distribution using the mean/variance model or the expected utility model. In the objective function, a feasibility penalty function is used to penalize violations of the control constraints under some of the scenarios, for example a quadratic penalty function or an exact penalty function. Therefore, the Mulvey's framework can be seen as a multi objective programming whose objectives are (1) the mean value, (2) the variability of the objective, and (3) penalty of violating control constraints. The framework has been applied to many problems such as the capacity expansion problem, the matrix-balancing problem, airline allocation for the air force, scenario immunization, and minimum weight structural design.

In this paper we propose and discuss an alternative model for aggregate production planning when periodic demands are uncertain. The objective of the model is to provide cost effective production plans while maintaining the targeted service levels. The model does not use scenarios explicitly which results in a huge savings in terms of variables and computational times.

2 Aggregate Production Planning Models

In this section, we present an aggregate production planning model when periodic demands are deterministic. We then present two production planning models when demands are uncertain, i.e. the scenario-based optimization model and an alternative new model. Let N be the number of products and T be the length of horizon planning. Let c_{it}^r and c_{it}^o be the production cost per unit product i during period t in regular time and in over time respectively. Let w_t^c, w_t^h, and w_t^l be respectively the labour, hiring and laying-off costs per worker during period t and w_t^{max}, h_t^{max}, l_t^{max} be the maximum available workforce, maximum hiring and maximum laying off in period t. Let f_{it} be the fixed cost of production for product i during period t and h_{it} be the holding cost per unit of product i by the end of period t. Let l_i and m_i be labour time and machine time needed per unit product i respectively. Let rwh be the working hour per labour per period. Let rmc_t be the regular machine capacity. Let omc_t and owc_t be the over time machine capacity and workforce (in fraction of regular capacity/workforce) respectively. Let I_{i0} and w_0 be the inventory level and the workforce level at the end of period 0, respectively. The variable of the problems are X_{it}^r and X_{it}^o which determine the quantity of production in period t during regular time and over time respectively. The variable Y_{it} is a binary variable which takes value 1 if production of product i takes place in period t. Let I_{it} be the inventory level for product i by the end of period t. Let W_t, WH_t and WL_t denote the workforce levels, the amount of workforce hired and the amount of workforce laid-off in

period t. Let d_{it} be the forecasted demand for product i in period t (in units), the deterministic aggregate production planning (DAPP) model is formulated as follow:

Minimize

$$\sum_{t=1}^{T}\sum_{i=1}^{N} f_{it}Y_{it} + c_{it}^{r}X_{it}^{r} + c_{it}^{o}X_{it}^{o} + h_{it}I_{it} + \sum_{t=1}^{T} w_t^c W_t + w_t^h WH_t + w_t^l WL_t \quad (1)$$

subject to

$$X_{it}^{r} + X_{it}^{o} + I_{it-1} - I_{it} = d_{it}, \forall i, t, \quad (2)$$

$$W_t = W_{t-1} + WH_t - WL_t, \forall t, \quad (3)$$

$$W_t \leq w_t^{max}, \forall t; WH_t \leq h_t^{max}, \forall t; WL_t \leq l_t^{max}, \forall t, \quad (4)$$

$$\sum_{i=1}^{N} l_i X_{it}^{r} \leq rwh\left(W_t\right), \forall t, \quad (5)$$

$$\sum_{i=1}^{N} l_i X_{it}^{o} \leq owc_t\left(rwh\right)\left(W_t\right), \forall t, \quad (6)$$

$$\sum_{i=1}^{N} m_i X_{it}^{r} \leq rwc_t; \sum_{i=1}^{N} m_i X_{it}^{o} \leq omc_t\left(rwc_t\right), \forall t, \quad (7)$$

$$m_i\left(X_{it}^{r} + X_{it}^{o}\right) \leq \left(1 + omc_t\right)\left(rmc_t\right) Y_{it}, \forall i, t, \quad (8)$$

$$X_{it}^{r}, X_{it}^{o}, I_{it}, W_t, WH_t, WL_t \geq 0, Y_{it} \in \{0, 1\}.$$

The total cost in the objective function (1) is the sum of all costs, i.e. fixed, production, holding, labour, hiring and layoff costs. The inventory balance and workforce balance are presented by constraints (2) and (3), respectively. Constraints (4) are the maximum availability for workforce, the maximum number of hiring and the maximum number of laid-off workforce. Constraints (5) and (6) show the capacity of labour hours for regular and overtime production. The capacity of machine is described by constraint (7), for both regular and over time capacity respectively. Constraint (8) shows the fixed cost realization, i.e. if production of i is taking place at period t. If the capacities of machines and labour are not sufficient to meet demand then the problem is infeasible (i.e. all demands must to be satisfied).

2.1 Scenario-Based Optimization Model

Leung *et al.* [6] proposed an aggregate production planning model based on the scenario realization as suggested by Mulvey *et al.* ([8]). Assume that the uncertain demand is represented by a set of scenario $s \in \Omega$ which is taking value d_{it}^{s} for product i in period t for scenario s with probability p^{s}. Assume

that the variables $X^r_{it}, X^o_{it}, W_t, WH_t, WL_t, Y_{it}$ are design variables which do not depend on the realization of the scenario and assume that $I^s_{i,t}$ are control variable which depend on the realization of the scenario $s \in \Omega$. We also consider the lost sales or unmet demand E^s_{it} with respect to infeasibility under scenario $s \in \Omega$. Constraints (2) then become control constraints as follow: $X^r_{it} + X^o_{it} + I^s_{it-1} - I^s_{it} + E^s_{it} = d^s_{it}, \forall i, t, \forall s \in \Omega$. Because the demand, inventory, and the unmet demand depend on the scenario realization, the cost function, ξ^s becomes a random variable taking value $\sum_{t=1}^{T}\sum_{i=1}^{N}\left(f_{it}Y_{it} + c^r_{it}X^r_{it} + c^o_{it}X^o_{it} + h_{it}I^s_{it}\right) + \sum_{t=1}^{T}\left(w^c_t W_t + w^h_t WH_t + w^l_t WL_t\right)$. The average cost is then defined as $\bar{\xi} = \sum_{s\in\Omega} p^s \xi^s$. The variability of the cost function can be measured using variance or mean absolute deviation. The variance model leads to a large number of computational times attributed to calculating the quadratic term. On the other hand, the mean absolute deviation can be easily transformed into linear form using equation $\xi^s - \bar{\xi} + \theta^s \geq 0$, where $\theta^s \geq 0$. The absolute deviation is then replaced by $\xi^s - \bar{\xi} + 2\theta^s \geq 0$ (for a complete discussion, see Leung *et al.* [6]). The unmet demand E^s_{it} is considered as a penalty in the objective function. The mean value of this penalty function is $\sum_{s\in\Omega}\sum_{t=1}^{T}\sum_{i=1}^{N} p^s E^s_{it}$. Using a constant ($\lambda$) times the mean absolute deviation as a choice of risk norm and the weight ω as the trade-off to feasibility robustness, the scenario-based aggregate production planning model is formulated as follow

Minimize

$$\sum_{s\in\Omega} p^s \xi^s + \lambda \sum_{s\in\Omega} p^s \left(\xi^s - \sum_{s'\in\Omega} p^{s'}\xi^{s'} + 2\theta^s\right) + \omega \sum_{s\in\Omega}\sum_{t=1}^{T}\sum_{i=1}^{N} p^s E^s_{it} \qquad (9)$$

subject to

$$X^r_{it} + X^o_{it} + I^s_{it-1} - I^s_{it} + E^s_{it} = d^s_{it}, \forall i, t, \forall s \in \Omega, \qquad (10)$$

$$\xi^s = \sum_{t=1}^{T}\sum_{i=1}^{N} f_{it}Y_{it} + c^r_{it}X^r_{it} + c^o_{it}X^o_{it} + h_{it}I^s_{it} + \sum_{t=1}^{T} w^c_t W_t + w^h_t WH_t + w^l_t WL_t, \qquad (11)$$

$$\xi^s - \sum_{s'\in\Omega} p^{s'}\xi^{s'} + \theta^s \geq 0, \forall s \in \Omega, \qquad (12)$$

$$X^r_{it}, X^o_{it}, E^s_{it}, I^s_{it}, W_t, WH_t, WL_t, \theta^s \geq 0, Y_{it} \in \{0,1\}, \forall i, t, \forall s \in \Omega.$$

Equation (9) shows the multi objective function which consists of the mean cost, weighted mean absolute deviation and weighted mean unmet demand. Constraints (10) are the inventory balance as a result of the scenario realization of the demand. Constraints (11) are the random value of the cost function, while constraints (12) are required for the transformation of the absolute deviation into a linear form. Constraints (3)-(8) are also required as in the deterministic model. The control constraints in Equation 10 suggests that infeasibility (unmet demand) under scenario s is acceptable but penalized by the parameter ω (i.e. it is considered to be a soft constraint). In this case, the production planner has to decide the level of 'acceptable' infeasibility in term of unmet demand.

2.2 An Alternative Model

The idea of generating a deterministic equivalent or approximation to the original stochastic problem has been initiated by Bitran & Yanasse ([2]). They showed that the relative error bound for the deterministic equivalent is small enough to justify its use for practical problems.

Our model is a deterministic equivalent which basically integrates the concept of safety stock into the aggregate production planning. The objective is to generate optimal plans that are robust, in sense that all realized demands are covered with a certain predetermined level of confidence. Assume that the periodic demands of product i, d_{it}, are stochastic, independent and are normally distributed $N(\bar{d_{it}}, \sigma_{it})$, where $\bar{d_{it}}$ is the average demand and σ_{it} is the standard deviation. For any consecutive $\{u, u+1, ..., v\}$ and $(u \leq v)$ set of period, the cumulative demand D_i^{uv} has a probability distribution with average $\bar{d_i^{uv}}$ and standard deviation σ_i^{uv} as follow $\bar{d_i^{uv}} = \sum_{\tau=u}^{v} \bar{d_{i\tau}}$ and $\sigma_i^{uv} = \sqrt{\sum_{\tau=u}^{v} \sigma_{i\tau}^2}$. Suppose that a $100(1-\alpha)\%$ service level is to be achieved then the plan must provides quantities at the beginning of period u to cover the realized demand from period u to v, $u \leq v$, such $P(D_i^{uv} \geq \bar{d_i^{uv}} + z_\alpha \sigma_i^{uv}) = \alpha$. The term $z_\alpha \sigma_i^{uv}$ refers to the safety stock required for product i to cover the variation of the demand from period u to period v, i.e. the interval between successive production, where z_α is a standard normal value. To model this, we introduce a new binary variable Z_i^{uv} defined for each pair period $(u, v), u \leq v$ which takes value 1 if production takes place at period u to cover integrally the realized demands from period u to period v, and zero otherwise. The alternative model can be then formulated as follows

Minimize

$$\sum_{t=1}^{T} \sum_{i=1}^{N} f_{it} Y_{it} + c_{it}^r X_{it}^r + c_{it}^o X_{it}^o + h_{it} I_{it} + \sum_{t=1}^{T} w_t^c W_t + w_t^h W H_t + w_t^l W L_t \quad (13)$$

subject to

$$X_{it}^r + X_{it}^o + I_{it-1} - I_{it} = \bar{d_{it}}, \forall i, t, \quad (14)$$

$$X_{it}^r + X_{it}^o + I_{it-1} \geq \left(\bar{d_i^{tv}} + z_\alpha \sigma_i^{tv}\right) Z_i^{tv}, \forall t \in T, t \leq v, \quad (15)$$

$$\sum_{u=1}^{t} \sum_{v=t}^{T} Z_i^{uv} = 1, \forall t \in T, t \leq v, \quad (16)$$

$$\sum_{v=t}^{T} Z_i^{tv} - Y_{it} \leq 0, \forall t \in T, \quad (17)$$

$$X_{it}^r, X_{it}^o, I_{it}, W_t, W H_t, W L_t \geq 0, Y_{it} \in \{0, 1\}, Z_i^{tv} binary, (t, v) \in T, t \leq v.$$

The objective function in the alternative model consists in minimizing the expected total cost, i.e. the production cost, expected holding cost (i.e. the inventory and safety stock), labour cost, hiring and layoff cost, and the fixed cost.

The inventory left from the previous period is unknown since the demand in the previous period is also stochastic. One way to model this is with a mean value approximation, assuming that demand is realized on the average. Constraints (14) are the expected inventory balance if the demands are realized on the average. Constraints (15) require that at the beginning of period t, production and inventory must be able to cover realized demand from period t to v, $(t \leq v) \in T$. Constraints (16) decompose the planning horizon T into a partition of subsets of consecutive periods. In equation (17), the variable Z_i^{tv} takes value 1 only if the variable Y_{it} also takes value 1, i.e. the production takes place at period t. The alternative model makes use the parameter z_α to determine the stock out probability which requires adding some safety stock between successive production to meet all realized demand for a certain level of confidence. Thus, the robustness in the alternative model for the aggregate production planning is simply defined by the ability of the plan to cope the variation of demand using an amount of safety stock.

3 Experimental Design and Computational Result

We evaluate both robust aggregate production planning models for 4-period problems comparing the impact of demand trends, the variability of demand, the capacity and the ratio of fixed cost and holding cost. We test the models for three demand trends, where the average demand is constant, increasing, and decreasing over time. We use the coefficient of variation $(\sigma_{it}/\bar{d_{it}})$ to measure the variability of the demand and assume that it increases over time as a result of insufficient information about the future demand. We test the models for two levels of the variability, i.e. low and medium variability (The coefficient of low variability is 0.1, 0.15, 0.2, and 0.25 for period 1,2,3, and 4 and 0.15, 0.25, 0.35, and 0.45 for the medium variability). We also evaluate the effect of the regular machine capacity levels which is assumed to be constant over time. Three different levels of capacity are evaluated, i.e. tight (110 %), medium (150 %) and loose (200 %) of the average demand in hours. The effect of the ratio of fixed costs over holding costs is also evaluated. We use different levels of the ratio, i.e. 0.5, 1, 2, and 4. We generate a number of scenarios to represent the stochastic demand for each period. The realization of demand for each period falls into three discrete values, i.e. $\left(\bar{d_{it}} - 2\sigma_{it}\right)$, $\left(\bar{d_{it}}\right)$, and $\left(\bar{d_{it}} + 2\sigma_{it}\right)$ with the probability 0.16, 0.68 and 0.16 respectively. The resulting probability density function approximates a normal distribution with average $\bar{d_{it}}$, and standard deviation σ_{it}. For an instance of problem (constant average demand, medium variability, medium capacity and the ratio of the fixed costs and the holding costs equals to 2), we solve the problem using the alternative model with the parameter $z_\alpha = 2$, i.e. we are expecting that the stock out probability is approximately 5 %.

Using the previously generated demand scenario, the performance of the plan is then calculated in term of mean cost (MC), mean absolute deviation (MAD), mean unmet demand (ME), and stock out probability (SO). The stock out probability is defined as $\sum_{s\in\Omega}\sum_{i=1}^{N}\sum_{t=1}^{T} p^s O_{it}^s/NT$, where O_{it}^s equals to 1 if $E_{it}^s > 0$,

and 0 otherwise. We also calculate the ratio between the mean absolute deviation and mean cost and use it as the constant λ for the scenario-based optimization model. We use the same λ for the scenario-based model because we want to compare both models in term of solution robustness (i.e. mean cost) and model robustness (i.e. mean unmet demand). The alternative model for such a case generates a plan where the ratio between the mean absolute deviation and the mean cost equals to 0.028. We use the same λ for different levels of the parameter weight ω for the scenario-based optimization model. Naturally, if $\omega = 0$ then the cost is minimum when no production takes place. However, this is not a feasible plan because the number of unmet demand, E_{it}^s is very high and the service level is 0%. To get a more feasible plan, we need to increase ω, i.e. giving more penalties for each unmet demand (infeasibility). However, having a more robust plan (in term of feasibility) comes with additional costs. The trade off between the cost and the expected number of unmet demand is shown in Figure 1. Using $\lambda = 0.028$ and different levels of ω, the production planners can decide which plan gives a 'good enough' expected cost with a 'reasonable' amount of unmet demand. The term 'good enough' and 'reasonable enough' are of course relatively subjective. If the production planner decides to use $\omega = 550$, the scenario-based optimization model generates a plan with mean cost equals to €543207 and mean unmet demand equals to 80 units. The plans and their performance measures generated from the alternative model and the scenario-based model are presented in Table (1).

To make a fair comparison in term of cost, we evaluate the plans which generate almost the same unmet demand for both models. Since the scenario-based optimization model depends on the parameter ω, we increment the parameter by 50 such that the unmet demand of the plan is approaching the unmet demand from the alternative plan. The computation results show that the demand

Table 1. Aggregate Production Plans

		Alternative model				Scenario-based $\lambda = 0.0283$, $\omega = 550$			
	Product	1	2	3	4	1	2	3	4
Regular production	1	560	766	820	727	154	2186	0	1493
	2	1560	1891	1970	2109	3449	0	3280	0
Overtime production	1	350	579	640	640	735	656	0	0
	2	0	0	0	0	1	0	984	0
Fixed cost realization	1	1	1	1	1	1	1	0	1
	2	1	1	1	1	1	0	1	0
Working labour		30	38	40	40	46	41	41	28
Hiring labour		0	8	2	0	16	0	0	0
Laying off		0	0	0	0	0	5	0	13
Mean cost (MC)					534988				543207
Abs. deviation (MAD)					15107				15392
Unmet demand (ME)					116				80
Stock out prob. (SO)					0.0308				0.0538
MAD/MC					0.0283				0.0283

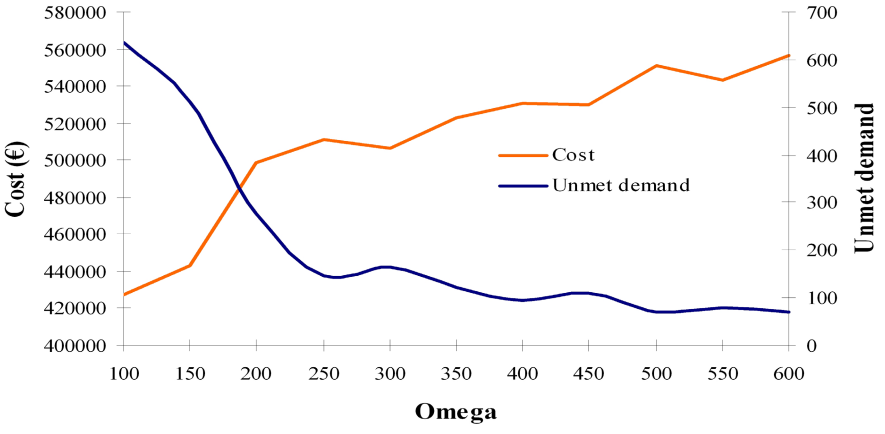


Fig. 1. Expected cost and unmet demand as function of ω

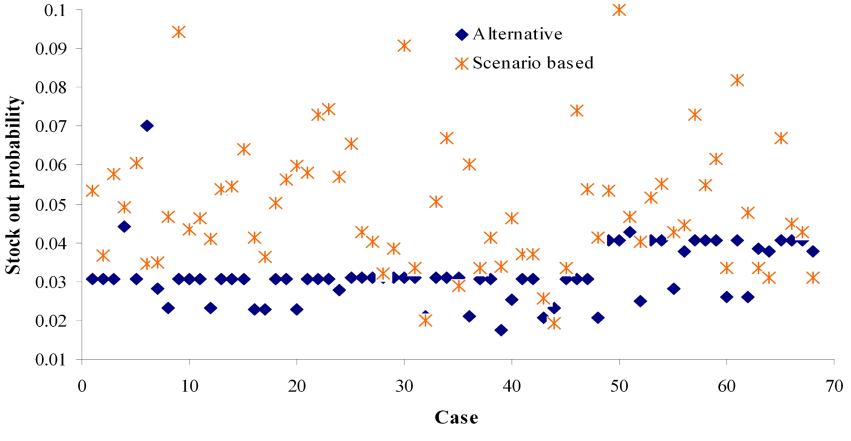


Fig. 2. Variation of stock out probability

trend does not influence the optimality of the plan for both models. Regardless of the capacity level, the variability influences the optimality for both models showing that if the variability is high then the cost is also high. In the alternative model, it corresponds to the need of a higher safety stock. The effect of capacity is noticeable when the ratio of fixed cost and holding cost is high. In this situation, the scenario-based model is superior when the capacity is tight. However, the difference becomes less noticeable when the capacity is loose. We do the same procedure to evaluate both models in term of stock out probability. Figure 2 shows that the alternative model generates plans that are more 'stable' in term of the stock-out probability. This results from the fact that the alternative model makes use of the parameter z_α which corresponds directly to the stock out probability. We evaluate the effect of the parameter z_α on the

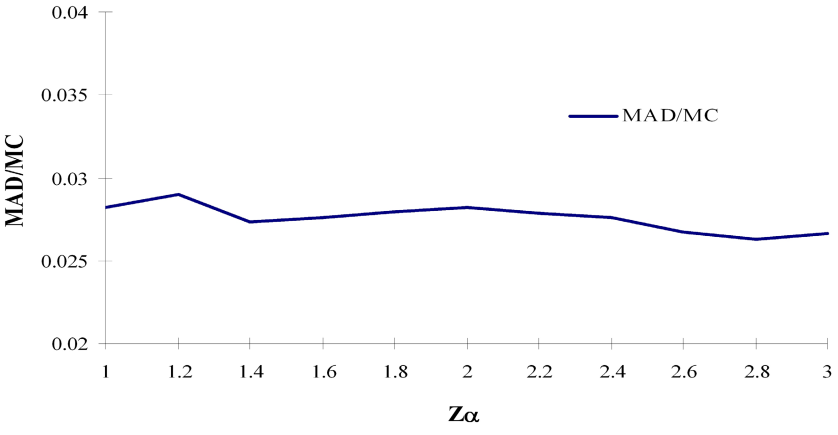


Fig. 3. Ratio of MAD/MC

optimality and the feasibility of the plan. We found that the trade off between the mean cost and the unmet demand as functions of the parameter z_α is similar to that of the scenario-based optimization model with parameter ω. Interestingly, the alternative model generates plans where the ratio between the variability of the cost (i.e. mean absolute deviation) and the mean cost bounded by 0.03 which means that the deviation is approximately 3 % from the mean cost (see Figure 3).

4 Conclusion and Further Research

The scenario-based optimization model and the alternative model generate plans that are capable of coping with the variability of demand in term of model robustness (i.e. expected unmet demand or the stock out level) and the solution robustness (i.e. the mean cost). The scenario-based optimization model makes use the parameter ω to penalize the unmet demand. The alternative model, on the other hand, makes use parameter z_α to assure a certain maximum level of the stock out probability. The trade off between the mean cost and the unmet demand can be established for both models using different levels of the parameters. In term of cost, the scenario-based model is only slightly superior when the capacity is tight and the ratio between the fixed cost and the holding cost is big. However, the alternative model needs less computational time than the scenario-based optimization model. In term of stock out probability, the alternative model provides plans with smaller and stable results than the scenario-based optimization model. The scenario-based optimization model is flexible in determining the level of variability of the cost from the mean cost using the parameter λ. In the other hand, the alternative model bounds the variability of the cost into 3 % of the mean cost. As mentioned previously, the robustness concept is system-specific. In term of production planning where the uncertainty comes

from the demand, safety stock can be considered as a tool to achieve a certain level of robustness (which is measured by the unmet demand or service level). The robustness approach for aggregate production planning dealing with uncertain demand in the alternative model is basically corresponding with the use of safety stock. The safety stock is represented by parameter z_α which determine the stock out probability. The alternative model which incorporates the safety stock, although more expensive in few cases, is able to produce plan that are robust in a reasonable computational time. The scenario-based optimization model, however, needs more computational time due to the number of constraint associated with the scenarios. However, this approach may not be sufficient when uncertainty appears also in other parameters such as costs and capacity. Thus, further research directed toward this types of situation is still needed.

References

1. Bai, D., Carpenter, T., Mulvey, J.M.: Making a case for robust optimization models. Management Science 43(7), 895–907 (1997)
2. Bitran, G.R., Yanasse, H.H.: Deterministic approximations to stochastic production problems. Operations Research 32(2), 999–1018 (1984)
3. Guide, V.D.R., Srivastaya, R.: A review of techniques for buffering against uncertainty with MRP systems. Production Planning & Control 11(3), 223–233 (2000)
4. Kouvelis, P., Yu, G.: Robust discrete optimization and its application. Kluwer Academic Publishers, Dordrecht (1997)
5. Lasserre, J.B., Bes, C., Roubellat, F.: The stochastic discrete dynamic lot size problem: an open loop solution. Operations Research 33(3), 684–689 (1985)
6. Leung, S.C.H., Wu, Y., Lai, K.K.: A robust optimization model for stochastic aggregate production planning. Production Planning & Control 15(5), 502–514 (2004)
7. List, G.F., Wood, B., Nowick, L.K., Turnquist, M.A., Jones, D.A., Kjeldgaard, E.A., Lawton, C.R.: Robust optimization for fleet planning under uncertainty. Transportation Research Part E 39(3), 209–227 (2003)
8. Mulvey, J.M., Vanderbei, R.J., Zenios, S.A.: Robust optimization of large-scale systems. Operations Research 43(2), 264–281 (1995)
9. Yu, C.-S., Li, H.-L.: A robust optimization model for stochastic logistic problems. International Journal of Production Economics 64, 385–397 (2000)
10. Yu, G.: Robust economic order quantity models. European Journal of Operational Research 100, 482–493 (1997)

Challenging the Incomparability Problem: An Approach Methodology Based on ZAPROS

Isabelle Tamanini and Plácido Rogério Pinheiro

University of Fortaleza (UNIFOR) - Master Course in Applied Computer Sciences
Av. Washington Soares, 1321 - Bl J Sl 30 - 60.811-905 - Fortaleza - Brazil
isabelle.tamanini@gmail.com, placido@unifor.br

Abstract. Aiming the reduction of incomparability cases between alternatives, for the presentation of a complete and satisfactory result, it is presented a new approach for aiding the decision making process on Verbal Decision Analysis, structured basically on ZAPROS III method. A tool applying the methodology was developed. Some optimizations were done to the method through some differentials on the process, considering similar tools which support solving ill-structured problems. Computational experiments applied to the tool presented promising results.

Keywords: Verbal Decision Analysis, ZAPROS, Multicriteria.

1 Introduction

One of the greatest problems faced on organizations is related to decision making process. The determination of the object which will conduct to the greatest result isn't a trivial process and involves series of factors to be analyzed. These problems are classified as complex and the consideration of all relevant aspects to the decision making is practically impossible, due to human limitations.

The decision making related to management decisions is a critical process, since the wrong choice between two alternatives can lead to a waste of resources, affecting the company. Complex problems found in organizations can be solved in a valid and complete way through the application of multicriteria methods, such as the work developed on the company Cascaju [7], which will be used as an application model for this paper.

2 ZAPROS III Method

The ZAPROS III method belongs to Verbal Decision Analysis (VDA) framework and it aims the classification of given multicriteria alternatives. The method is structured on the acknowledgment that most of the decision making problems can be verbally described. The Verbal Decision Analysis supports the decision making process by verbal representation of the problem [3].

The method is based on elicitation of preferences around values that represent distances between evaluations of two criteria. A scale of preferences can be structured, enabling the comparison of alternatives.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 338–347, 2008.

Before the alternatives comparison process, one should consider:

- The preferences must be elicited such that a decision rule can be formed before the presentation of alternatives;
- The comparisons between criteria will be made by human beings, symbolizing the decision maker (DM);
- The quality graduations of criteria values are verbal and defined by the DM.

Among the advantages of ZAPROS III method utilization, we can say that [9]:

- It presents questions on elicitation of preferences process understandable to the decision maker, based on criteria values. This procedure is psychologically valid (because it respects the limitations of the human information processing system) and represents the method's greatest feature;
- It presents considerable resistance to decision maker's contradictory inputs, being capable of detect and request a solution to these problems;
- It specifies all informations of qualitative comparison on the decision maker's natural language.

A disadvantage of the method is that the number of criteria and values of criteria supported are limited, since they are responsible for the exponential growth of the problem alternatives and of the information required on the process of preferences elicitation.

The scale of preferences is essentially qualitative, defined with verbal variables, causing losses on the comparison power, because these symbols aren't assigned of exact values (which implies in the inexistence of overall values - best or worst in any kind of situation) and can't be recognized computationally. So, there are a lot of incomparable alternatives, what can lead to an absence of an acceptable result.

According to [5], the estimative of the incomparable alternatives number (and, consequently, of the method's decision power) can be made by calculating the number of pairs of alternatives ($Q = 0.5n^N(n^N - 1)$, where N represents the number of criteria and n is the number of criteria values) and the subset that will be related by Pareto's dominance (D). From the difference between Q and D, we have the set of alternatives that depends directly of the preferences' scale obtained by the decision maker's answers, this is the set with the greatest probability of presenting contradictory pairs of alternatives. After that, the index of decision power of the method can be obtained as follows: P=1-S/B, where B is the difference between Q and D, and S is the number of alternatives that can't be compared based on the DM's scale of preferences (incomparable alternatives).

On [2], a system implemented in Visual C++ 6.0 structured on ZAPROS III method is presented with an analysis of the method's performance. The Formal Index of Quality (FIQ), which allows the reduction of the comparisons number between alternatives at the process of alternatives' classification, is not used on the system.

The system presented on [1], called UniComBOS, aims avoiding the existing limitations of other methods, besides modifying the rule of consistency control of the decision maker's answers in order to improve it by utilization of procedures

beyond the transitivity relations. It is the implementation of a new procedure for comparison and classification of multicriteria alternatives.

The questions made at the process of preferences' elicitation involve only the criteria values necessary to compare one alternative to another (which is more preferable, which are equivalent, etc.). After the elicitation process, the user can check the solution proposed by the tool.

As the tool is based on preferences' elicitation only after the alternatives' definition, there is no decision rule formulated previously, instead of what occurs on ZAPROS method. An advantage of the implementation is that it can avoid the incomparability cases, but if a new alternative is defined or changed, the scale of preferences will be reevaluated and, possibly, modified. On simulation scenarios, where objects will be constantly modified, or in cases where there is no direct access to the decision maker or to the alternatives, as in a model of decision making for computational agents, that would not be an indicated tool.

3 A New Approach Methodology

A methodology structured basically on ZAPROS III method [4] is proposed. It presents three main stages: *Problem's Formulation*, *Elicitation of Preferences* and *Comparison of Alternatives*, as proposed on the original version of the ZAPROS method. These stages are described as follows.

3.1 Formal Statement of the Problem

The methodology follows the same problem's formulation proposed on [4]:

Given:

1) K = 1, 2,..., N, representing a set of N criteria;

2) n_q represents the number of possible values on the scale of q-th criterion, $(q \in K)$; for the ill-structured problems, as in this case, usually $n_q \leq 4$;

3) $X_q = x_{iq}$ represents a set of values to the q-th criterion, which is this criterion scale; $|X_q| = n_q (q \in K)$; where the values of the scale are ordered from best to worst, and this order doesn't depends on the values of other scales;

4) $Y = X_1 * X_2 * ... * X_n$ represents a set of vectors y_i, such that: $y_i = (y_{i1}, y_{i2}, ..., y_{iN})$, and $y_i \in Y$, $y_{iq} \in X_q$ and $P = |Y|$, where $|Y| = \prod_{i=1}^{i=N} n_i$.

5) $A = \{a_i\} \in Y$, i=1,2,...,t, where the set of t vectors represents the description of the real alternatives.

Required: The multicriteria alternatives classification based on the decision maker's preferences.

3.2 Elicitation of Preferences

In this stage, the scale of preferences for quality variations (Joint Scale of Quality Variations - JSQV) is constructed. The methodology follows the order of steps shown on fig. 1. This structure is the same proposed on [4], however, the substages 2 and 3 (numbered on the left side of the figure) were put together in just one substage.

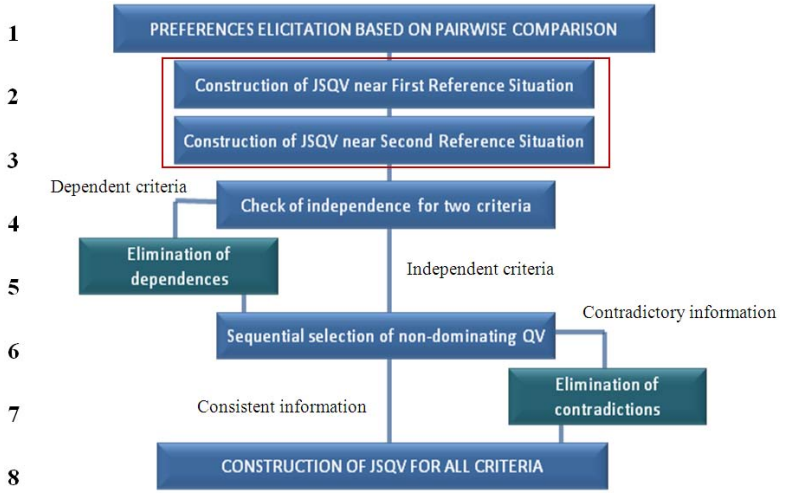


Fig. 1. Elicitation of preferences process

Instead of setting the decision maker's preferences based on the first reference situation and, then, establish another scale of preferences using the second reference situation, it is proposed that the two substages be transformed in one. The questions made considering the first reference situation are the same that the ones made considering the second reference situation. So, both situations will be presented and must be considered on the answer to the question, in order to not cause dependence of criteria. The alteration reflects on an optimization of the process: instead of making 2*n questions, only n will be made. The questions to quality variations (QV) belonging to just one criteria will be made as follows: supposing a criterion A with three values $X_A = A_1, A_2, A_3$, the decision maker will be asked about his preferences between the QV $a_1 - a_2$, $a_1 - a_3$ and $a_2 - a_3$. Thus, there is a maximum of three questions to a criterion with three values ($n_q = 3$).

The question will be formulated in a different way on the preferences elicitation for two criteria, because it was observed difficulties on understanding and delay on the decision maker's answers when exposing QV of different criteria.

The question will be made dividing the QV in two items. For example, having the set of criteria k = A, B, C, where $n_q = 3$ and $X_q = q_1, q_2, q_3$. Considering the pair of criteria A, B and the QV a_1 and b_1, the decision maker should analyze which imaginary alternative would be preferable: A_1, B_2, C_1 or A_2, B_1, C_1. However, this answer must be the same to the alternatives A_1, B_2, C_3 and A_2, B_1, C_3. If the decision maker answers that the first option is better, then b_1 is preferable to a_1, because it is preferable to have the value B_2 on the alternative instead of A_2.

3.3 Comparison of Alternatives

With the aim of reducing the incomparability cases, we apply the same structure proposed on [4], but modifying the comparison of pairs of alternatives' substage according to the one proposed on [6].

Each alternative has a function of quality - V(y) [4], depending on the evaluations on criteria that it represents. On [6], it is proposed that the vectors of ranks of criteria values, which represent the function of quality, are rearranged in ascending order. Then, the values will be compared to the corresponding position of another alternative's vector of values based on Pareto's dominance rule. Meanwhile, this procedure was modified to implementation because it was originally proposed to scales of preferences of criteria values, not for quality variations' scales.

So, supposing the comparison between alternatives $Alt1 = A_2, B_2, C_1$ and $Alt2 = A_3, B_1, C_2$, considering a scale of preferences: $a_1 \prec b_1 \prec c_1 \prec a_2 \prec b_2 \prec c_2 \prec a_3 \prec b_3 \prec c_3$, we have the following functions of quality: V(Alt1) = (0, 0, 2) and V(Alt2) = (0, 3, 4), which represents the ranks of, respectively, b_1 and c_1, a_2. Comparing the ranks presented, we can say that Alt1 is preferable to Alt2.

However, there are cases in which the incomparability of real alternatives won't permit a presentation of a complete result. These problems require a further comparison.

In such cases, we can evaluate all possible alternatives to the problem in order to rank indirectly the real alternatives. The possible alternatives should be rearranged in ascending order according to their Formal Index of Quality (FIQ) and only the significant part will be selected to the comparison process (the set of alternatives presenting FIQ between the greater and the smaller real alternatives' FIQ). After that, the ranks obtained will be passed to the corresponding real alternatives.

4 Proposed Tool

In order to facilitate the decision process and perform it consistently, observing its complexity and with the aim of turning it accessible, we present a tool implemented in Java, structured on Verbal Decision Analysis, considering the new approach methodology.

The tool is presented by the sequence of actions that follows:

- *Criteria Definition:* First of all, the user should define the criteria presented by the problem. In this stage occurs the problem formulation.

- *Preferences Elicitation:* This process occurs in two stages: elicitation of preferences for quality variation on the same criteria and elicitation of preferences between pairs of criteria.

- *Alternatives Definition:* The alternatives can be defined only after the construction of the scale of preferences.

- *Alternatives Classification:* After the problem formulation, the user can verify the solution obtained to the problem. The result is presented to the decision maker so that it can be evaluated.

If the classification of the alternatives isn't satisfactory, the decision maker can request a new comparison based on all possible alternatives for the problem. This is an elevated cost solution and should be performed only when it's necessary to the problem resolution.

5 Application of the Tool

We present the computational experiments applied to the tool. These were based on experiments with results already determined. The problem, its original result and the one obtained with the tool will be exposed.

5.1 Industrialization Process of Cashew Chestnut

The tool was submitted to the problem "A Model Multicriteria Applied to the Industrialization Process of the Cashew Chestnut" [7]. The choice of the more indicated industrialization process applied to cashew chestnut so that, after series of steps, one obtains a good index of whole almonds, involves the analysis of seven stages (formulated as criteria and presented in table 1). The decision was taken observing historical data and the tacit experience of the manager. The problem was modeled as a VDA problem and the scale of preferences was formulated according to the manager's (DM) informations.

The alternatives and their criteria representations, the original ranks and the ones obtained with the tool are exposed on table 2. The results presentation screen is exposed on fig. 2.

The application [7] resulted in a non satisfactory classification when applying purely the ZAPROS III method because of the incomparability cases; thus, the FIQ was used in the paper to rank order the alternatives. The proposed method rank ordered the given alternatives through pairwise comparison of all possible alternatives. The results obtained, although, were the same as the ordered given by FIQ.

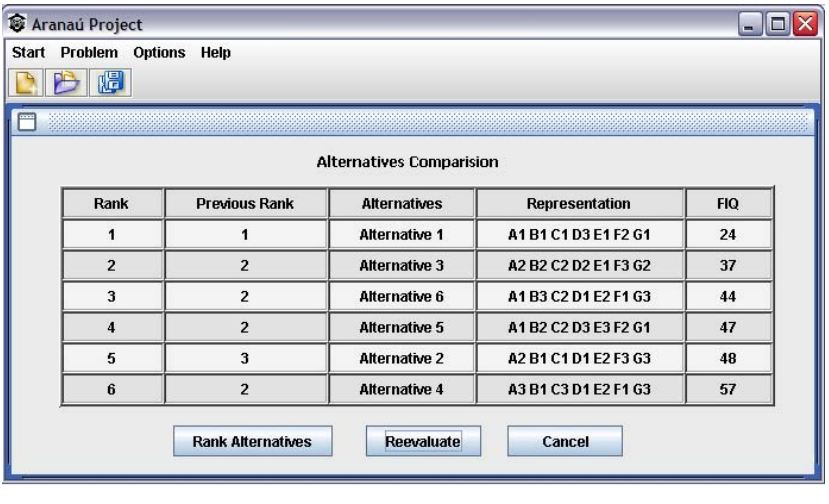


Fig. 2. Presentation of results

Table 1. Criteria involved on the cashew chestnut industrialization process

Criteria	Values of Criteria
A - Immersion Time	A1. 0 - 40 minutes
	A2. 41 - 80 minutes
	A3. 81 - 120 minutes
B - Rest Time in the Water	B1. 50 - 57 hours
	B2. 58 - 65 hours
	B3. 66 - 72 hours
C - Humidity Tax	C1. 8,90 - 10,0
	C2. 10,1 - 12,4
	C3. 12,5 - 13,0
D - Baking Temperature	D1. 180 C - 198 C
	D2. 199 C - 216 C
	D3. 217 C - 235 C
E - LCC Viscosity	E1. 150 cps - 334 cps
	E2. 335 cps - 520 cps
	E3. 521 cps - 700 cps
F - Entrance Outflow	F1. 800 kg/h - 966 kg/h
	F2. 967 kg/h - 1.133 kg/h
	F3. 1.134 kg/h - 1.300 kg/h
G - Cooling Temperature	G1. 38 C - 45 C
	G2. 46 C - 53 C
	G3. 54 C - 60 C

Table 2. Alternatives classifications of cashew chestnut industrialization process

Alternatives	Evaluations on Criteria	Original Rank	Rank Obtained
Alternative 1	A1B1C1D3E1F2G1	1	1
Alternative 2	A2B1C1D1E2F3G3	5	5
Alternative 3	A2B2C2D2E1F3G2	2	2
Alternative 4	A3B1C3D1E2F1G3	6	6
Alternative 5	A1B2C2D3E3F2G1	4	4
Alternative 6	A1B3C4D1E2F1G3	3	3

5.2 The Choice of a Prototype for Digital Mobile Television

The tool was also submitted to the problem presented on [8]. Three prototypes of mobile interfaces are evaluated according to user's opinion after using each one. The problem was formulated as a VDA problem and the information obtained was transformed into a scale of preferences.

The relevant criteria and their possible values are listed in table 3.

The problem was applied to the tool and presented the results exposed in table 4. The application also used the FIQ to rank order the alternatives. The

Table 3. Criteria involved on choosing a prototype for digital mobile television

Criteria	Values of Criteria
A - Functions Evidence	A1. No difficulty was found on identifying the system functionalities; A2. Some difficulty was found on identifying the system functionalities; A3. It was hard to identify the system functionalities.
B - User's familiarity with a determined technology	B1. No familiarity is required with similar applications of a determined technology; B2. Requires little user familiarity with applications of a determined technology; B3. The manipulation of the prototype is fairly easy when the user is familiar with similar applications.
C - User's locomotion while manipulating the device	C1. The user was not hindered in any way when manipulating the prototype while moving; C2. The user was occasionally confused when manipulating the prototype while moving; C3. The spatial orientation of the application was hindered when the user was moving.
D - Content Influence	D1. There is no influence of content on choosing the interface; D2. The content exerted some influence on choosing the interface; D3. The content was decisive on choosing the interface.
E - User Emotion	E1. He felt fine (modern, comfortable) when using the interface; E2. He felt indifferent when using the interface; E3. He felt bad (uncomfortable, frustrated) when using the interface.

Table 4. Alternatives classifications of prototypes for digital mobile television

Alternatives	Evaluations on Criteria	Original Rank	Rank Obtained
Prototype 1	A2 B1 C2 D1 E2	3	2
Prototype 2	A2 B3 C1 D1 E1	2	2
Prototype 3	A2 B1 C1 D1 E2	1	1

new methodology application resulted in the same classification order exposed on the original problem, which corresponds to the FIQ order.

Fig. 3 presents the preferences elicitation of quality variations to pairs of criteria.

An advantage of the ZAPROS III method is the presentation of all questions during the elicitation of preferences process in the decision maker's natural language, respecting the human information processing system limitations. The incomparability cases, however, are unavoidable when the scale of preferences is purely verbal, because there is no exact measure of the values.

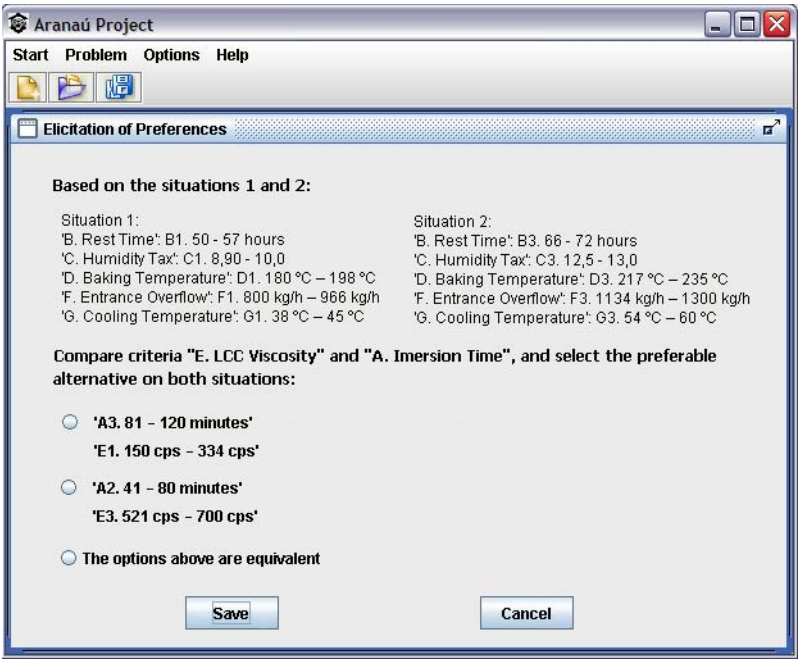


Fig. 3. Elicitation of preferences for two criteria - Application [7]

6 Conclusions

This paper contribution is the new approach methodology structured basically on ZAPROS III method and with some modifications in order to improve the alternatives comparison process. This methodology is presented by the tool, which allows aiding the decision making process.

In such case, it was used the Formal Index of Quality (FIQ) [4], which purpose is to reduce the number of pairs of alternatives compared; the ideas of comparison between alternatives by ordering the quality vectors in ascending order [6]; and, for a complex decision making process, it is possible to perform a comparison based on all possible alternatives for the problem.

These steps were sufficient to guarantee the comparison of all alternatives for the presented experiments.

As future works, we intend to improve the treatment proposed to incomparability cases, reducing the complexity it represents and, consequently, the execution cost of the procedure. New experiments on health areas will be done, aiming advances on early diagnosis of diseases.

As a conclusion to the structuring of the tool, the expected behavior was obtained by presentation of complete and satisfactory results at the end of the execution.

Acknowledgment. The authors are thankful to Celestica do Brasil and CNPq for the support received on this project.

References

1. Ashikhmin, I., Furems, E.: UniComBOS - Intellectual Decision Support System for Multicriteria Comparison and Choice. Journal of Multicriteria Decision Analysis 13(2), 147–157 (2005)
2. Dimitriadi, G.G., Larichev, O.: Decision Support System and the ZAPROS III Method for Ranking the Multiattribute Alternatives with Verbal Quality Estimates. Automation and Remote Control 66(8), 1322–1335 (2005)
3. Larichev, O., Moshkovich, H.M.: Verbal decision analysis for unstructured problems. Kluwer Academic Publishers, Boston (1997)
4. Larichev, O.: Ranking Multicriteria Alternatives: The Method ZAPROS III. European Journal of Operational Research 131(3), 550–558 (2001)
5. Larichev, O.: Method ZAPROS for Multicriteria Alternatives Ranking and the Problem of Incomparability. Informatica 12(1), 89–100 (2001)
6. Moshkovich, H., Mechitov, A., Olson, D.: Ordinal Judgments in Multiattribute Decision Analysis. European Journal of Operational Research 137(3), 625–641 (2002)
7. Carvalho, A.L., de Castro, A.K.A., Pinheiro, P.R., Rodrigues, M.M., Gomes, L.F.A.M.: Model Multicriteria Applied to the Industrialization Process of the Cashew Chestnut. In: 3rd IEEE International Conference Service System and Service Management, pp. 878–882. IEEE Press, New York (2006)
8. Tamanini, I., Machado, T.C.S., Mendes, M.S., Carvalho, A.L., Furtado, M.E.S., Pinheiro, P.R.: A model for mobile television applications based on verbal decision analysis. In: Elleithy, K. (ed.) Advances and Innovations in Systems, Computing Sciences and Software Engineering, vol. 19. Springer, Heidelberg (to appear, 2008)
9. Ustinovich, L., Kochin, D.: Verbal Decision Analysis Methods for Determining the Efficiency of Investments in Construction. Foundations of Civil and Environmental Engineering 5(1), 35–46 (2004)

DC Programming Approach for a Class of Nonconvex Programs Involving l_0 Norm

Mamadou Thiao[1], Tao Pham Dinh[1], and Hoai An Le Thi[2]

[1] Laboratory of Modelling, Optimization and Operations Research,
LMI, National Institute for Applied Sciences - Rouen
BP 08, Place Emile Blondel F 76131 Mont Saint Aignan Cedex, France
thiaoma@insa-rouen.fr, pham@insa-rouen.fr
[2] Laboratory of Theoretical and Applied Computer Science,
UFR MIM, Metz University, Ile du Saulcy, 57045 Metz, France
lethi@univ-metz.fr

Abstract. We propose a new solution for a class of nonconvex programs involving l_0 norm. Our method is based on a reformulation of these programs as bilevel programs, in which the objective function in the first level program is a DC function, and the second level consists of finding a Karush-Kuhn-Tucker point of a linear programming problem. Exact penalty techniques are then used to reformulate the obtained programs as DC programs. The resulted problems are then handled by the local algorithm DCA in DC programming. Preliminary computational results are reported.

Keywords: l_0 norm, bilevel programming, nonconvex programming, DC programming, DCA.

1 Introduction

We propose exact reformulations of some programs involving l_0 norm. Let

$$\|x\|_0 = |\{i \in \{1, ..., n\} : x_i \neq 0\}|.$$

Consider the problems :

$$\begin{cases} \min_{(x,y)} & f(x,y) \\ s.t. & \|x\|_0 \leq k, \\ & (x,y) \in K, \end{cases} \tag{1}$$

and

$$\begin{cases} \min_{(x,y)} & f(x,y) + \rho \|x\|_0 \\ s.t. & (x,y) \in K, \end{cases} \tag{2}$$

where K is a compact convex polyhedral subset of $\mathbb{R}^n \times \mathbb{R}^m$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a finite DC function, $\rho > 0$, and $1 \leq k < n$ are given.

The class of programs (1) and (2) are directly related to some optimization problems in learning (see Schnörr et al. [2] and Weston et al. [1] for example).

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 348–357, 2008.

They have many uses in a machine learning context: for variable or feature selection, minimizing the training error and ensuring the sparsity in solutions.

In this paper we first reformulate (1) and (2) as bilevel programs, in which the objective function in the first level is a DC function, and the second level consists of finding a Karush-Kuhn-Tucker point of a linear programming problem. Then we apply the theory of exact penalization of mathematical programs with equilibrium constraints developed by Luo et al [6] and the exact penalty techniques in DC programming due to Le Thi et al [4], [5] to reformulate the obtained bilevel programs as problems of minimizing DC functions over polyhedral convex sets. The resulted problems are then handled by a local approach in DC programming developed by Pham Dinh and Le Thi in their early works (see [4], [5], [7]).

The paper is organized as follows. In sections 2 and 3 we reformulate the problems (1) and (2) as DC Programs. The algorithm to solve the obtained problem is presented in section 3 while some applications in feature selection and numerical results are presented in section 4 and 5.

The proofs of propositions and lemmas were omitted for reasons of limitation of pages, refer to the technical report [12] for more details.

2 Reformulations

2.1 First Reformulation

Consider the following optimization problems

$$\begin{cases} \min_{(x,y,u,v)} & f(x,y) \\ s.t. & (x,y) \in K, \langle e_n, u+v\rangle \leq k, \\ & u \in \operatorname{argmin}\left\{\langle -x, \tilde{u}\rangle : \tilde{u} \in [0,1]^n\right\}, \\ & v \in \operatorname{argmin}\left\{\langle x, \tilde{v}\rangle : \tilde{v} \in [0,1]^n\right\}, \end{cases} \tag{3}$$

and

$$\begin{cases} \min_{(x,y,u,v)} & f(x,y) + \rho\langle e_n, u+v\rangle \\ s.t. & (x,y) \in K, \\ & u \in \operatorname{argmin}\left\{\langle -x, \tilde{u}\rangle : \tilde{u} \in [0,1]^n\right\}, \\ & v \in \operatorname{argmin}\left\{\langle x, \tilde{v}\rangle : \tilde{v} \in [0,1]^n\right\}, \end{cases} \tag{4}$$

where $e_n = (1, ..., 1)^T \in \mathbb{R}^n$.

Proposition 1. *1. The problems* (1) *and* (3) *are equivalent in the following sense :*

- *if* $(\bar{x}, \bar{y})$ *is an optimal solution of* (1)*, then there exists* $\bar{u}$ *and* $\bar{v}$ *such that* $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ *is an optimal solution of* (3)*, and*
- *if* $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ *is an optimal solution of* (3)*, then* $(\bar{x}, \bar{y})$ *is an optimal solution of* (1)*.*

2. The problems (2) *and* (4) *are equivalent in the following sense:*

- *if* $(\bar{x}, \bar{y})$ *is an optimal solution of* (2)*, then there exists* $\bar{u}$ *and* $\bar{v}$ *such that* $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ *is an optimal solution of* (4)*, and*
- *if* $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ *is an optimal solution of* (4)*, then* $(\bar{x}, \bar{y})$ *is an optimal solution of* (2)*.*

2.2 Second Reformulation

Applying the KKT conditions to (3) and (4), respectively we get the following problems

$$\begin{cases} \min_{(y,u,v,\lambda,\alpha)} & f(\lambda-\alpha,y) \\ s.t. & (\lambda-\alpha,y)\in K, \langle e_n, u+v\rangle \le k \\ & \langle \lambda, e_n-u+v\rangle = 0, \langle \alpha, e_n+u-v\rangle = 0 \\ & u,v\in[0,1]^n, \lambda,\alpha\in\mathbb{R}^n_+, \end{cases} \tag{5}$$

and

$$\begin{cases} \min_{(y,u,v,\lambda,\alpha)} & f(\lambda-\alpha,y)+\rho\langle e_n, u+v\rangle \\ s.t. & (\lambda-\alpha,y)\in K \\ & \langle \lambda, e_n-u+v\rangle = 0, \langle \alpha, e_n+u-v\rangle = 0 \\ & u,v\in[0,1]^n, \lambda,\alpha\in\mathbb{R}^n_+. \end{cases} \tag{6}$$

The connection between (5), (6) and (3), (4) can be described as follows.

Proposition 2. *1. The problems* (3) *and* (5) *are equivalent in the following sense :*
- *if* $(\bar{x},\bar{y},\bar{u},\bar{v})$ *is an optimal solution of* (3)*, then there exist* $\bar{\lambda}$*, and* $\bar{\alpha}$ *such that* $(\bar{y},\bar{u},\bar{v},\bar{\lambda},\bar{\alpha})$ *is an optimal solution of* (5)*, and*
- *if* $(\bar{y},\bar{u},\bar{v},\bar{\lambda},\bar{\alpha})$ *is an optimal solution of* (5)*, then* $(\bar{\lambda}-\bar{\alpha},\bar{y},\bar{u},\bar{v})$ *is an optimal solution of* (3)*.*

2. The problems (4) *and* (6) *are equivalent in the following sense:*
- *if* $(\bar{x},\bar{y},\bar{u},\bar{v})$ *is an optimal solution of* (4)*, then there exist* $\bar{\lambda}$*, and* $\bar{\alpha}$ *such that* $(\bar{y},\bar{u},\bar{v},\bar{\lambda},\bar{\alpha})$ *is an optimal solution of* (6)*, and*
- *if* $(\bar{y},\bar{u},\bar{v},\bar{\lambda},\bar{\alpha})$ *is an optimal solution of* (6)*, then* $(\bar{\lambda}-\bar{\alpha},\bar{y},\bar{u},\bar{v})$ *is an optimal solution of* (4)*.*

3 DC Programming and DCA for (5) and (6)

First of all, to make the paper self-contained and so more comprehensive for the reader not familiar with DC programming and DCA, we will outline main theoretical and algorithmic results on the topic.

3.1 DC Programming and DCA

Let $\Gamma_0(\mathbb{R}^n)$ denote the convex cone of all lower semicontinuous proper convex functions on $\mathbb{R}^n$. The vector space of DC functions, $DC(\mathbb{R}^n)=\Gamma_0(\mathbb{R}^n)-\Gamma_0(\mathbb{R}^n)$, is quite large to contain almost real life objective functions and is closed under all the operations usually considered in Optimization. Consider the standard DC program

$$(P_{dc}) \qquad \alpha = \inf\{f(x) := g(x)-h(x) : x\in\mathbb{R}^n\},$$

where $g,h\in\Gamma_0(\mathbb{R}^n)$. Remark that the closed convex constraint set C is incorporated in the first convex DC component g with the help of its indicator function $\chi_C(\chi_C(x)=0$ if $x\in C,+\infty$ otherwise).

Based on local optimality conditions and duality in DC programming, the DCA consists in the construction of two sequences x^k and y^k (candidates to be solutions of (P_{dc}) and (D_{dc}) resp.). Each iteration of DCA approximates the concave part $-h$ by its affine majorization (that corresponds to taking $y^k \in \partial h(x^k)$) and minimizes the resulting convex function (that is equivalent to determining $x^{k+1} \in \partial g^*(y^k)$).

Generic DCA scheme:
Initialization Let $x^0 \in \mathbb{R}^n$ be a best guest, $0 \leftarrow k$.
Repeat
Calculate $y^k \in \partial h(x^k)$.
Calculate $x^{k+1} \in \operatorname{argmin}\{g(x) - h(x^k) - \langle x - x^k, y^k\rangle : x \in \mathbb{R}^n\}$ (P_k).
$k+1 \leftarrow k$.
Until convergence of x^k.

Convergence properties of DCA and its theoretical basis can be found in Pham Dinh, Le Thi et al. [4], [5], and [7].

DCA's Convergence Theorem. DCA is a descent method without linesearch which enjoys the following primal properties (the dual ones can be formulated in a similar way):

1. The sequences $\{g(x^k) - h(x^k)\}$ and $h^*(y^k) - g^*(y^k)$ are decreasing.
2. If the optimal value α of problem (P_{dc}) is finite and the infinite sequences $\{x^k\}$ and $\{y^k\}$ are bounded then every limit point x^∞ (resp. y^∞) of the sequence $\{x^k\}$ (resp. $\{y^k\}$) is a critical point of $g - h$ (resp. $h^* - g^*$).
3. DCA has a linear convergence for general DC programs.
4. For polyhedral DC programs, DCA has a finite convergence.

We shall apply *all DC enhancement features* to solve (1) and (2).

3.2 DCA for Solving (1) and (2)

In this section we will reformulate (5) and (6) as equivalent DC programs, where f is a DC function and K is a compact convex polyhedral subset of $\mathbb{R}^n \times \mathbb{R}^m$, and then apply DCA. We emphasize the finite convergence of DCA and its feasibility in the related polyhedral DC programs.

Reformulation of the Penalty Equivalents as DC Programs Let

$$q(y,u,v,\lambda,\alpha) := \sum_{l=1}^{n} (min(\lambda_l, 1 - u_l + v_l) + min(\alpha_l, 1 + u_l - v_l)), \tag{7}$$

and

$$\Omega := \{(y,u,v,\lambda,\alpha) : (\lambda - \alpha, y) \in K, u, v \in [0,1]^n, \lambda, \alpha \in \mathbb{R}^n_+\}. \tag{8}$$

Property 1. 1. q is finite nonnegative concave in Ω.
2. $-q$ is convex polyhedral.

As K is bounded, there exists $M > 0$ such that $|x_l| \leq M$ for all $l \in \{1, ..., n\}$ and for all $(x, y) \in K$. Let

$$\tilde{\Omega} := \{(y, u, v, \lambda, \alpha) : (\lambda - \alpha, y) \in K, u, v \in [0, 1]^n, \lambda, \alpha \in [0, M + 2]^n\}. \quad (9)$$

Lemma 1. *1. Ω is nonempty closed convex polyhedral set.*
2. $\{(y, u, v, \lambda, \alpha) : (y, u, v, \lambda, \alpha) \in \Omega, q(y, u, v, \lambda, \alpha) \leq 0\} \subset \tilde{\Omega} \subset \Omega$.

Based on this lemma, we can rewrite (5) and (6), respectively, as

$$\begin{cases} \min_{(y,u,v,\lambda,\alpha)} & f(\lambda - \alpha, y) \\ s.t. & \langle e_n, u + v\rangle \leq k, q(y, u, v, \lambda, \alpha) \leq 0, (y, u, v, \lambda, \alpha) \in \tilde{\Omega}, \end{cases} \quad (10)$$

and

$$\begin{cases} \min_{(y,u,v,\lambda,\alpha)} & f(\lambda - \alpha, y) + \rho\langle e_n, u + v\rangle \\ s.t. & q(y, u, v, \lambda, \alpha) \leq 0, (y, u, v, \lambda, \alpha) \in \tilde{\Omega}. \end{cases} \quad (11)$$

Let $t > 0$. Consider the following penalty programs

$$\begin{cases} \min_{(y,u,v,\lambda,\alpha)} & f(\lambda - \alpha, y) + tq(y, u, v, \lambda, \alpha) \\ s.t. & \langle e_n, u + v\rangle \leq k, (y, u, v, \lambda, \alpha) \in \tilde{\Omega}, \end{cases} \quad (12)$$

and

$$\begin{cases} \min_{(y,u,v,\lambda,\alpha)} & f(\lambda - \alpha, y) + \rho\langle e_n, u + v\rangle + tq(y, u, v, \lambda, \alpha) \\ s.t. & (y, u, v, \lambda, \alpha) \in \tilde{\Omega}. \end{cases} \quad (13)$$

Proposition 3. *1. There exists $t_1 > 0$ such that* (5) *and* (12) *are equivalent for all $t > t_1$.*
2. There exists $t_2 > 0$ such that (6) *and* (13) *are equivalent for all $t > t_2$.*

The objective functions of (12) and (13) are nondifferentiable and nonconvex. They are actually DC functions, so (12) and (13) are DC programs. Note that if f is a convex or DC function whose first DC component is polyhedral convex, then (12) and (13) are polyhedral DC programs for which DCA has a finite convergence.

First we have to present (12) and (13) in the standard form of a DC program. Since the function f is DC (with respect to the pair of variables (x, y)) on K

$$f(x, y) = f_1(x, y) - f_2(x, y),$$

with f_1 and f_2 being convex functions on K. The function F defined by

$$F(y, u, v, \lambda, \alpha) = f(\lambda - \alpha, y)$$

is DC with the following DC decomposition

$$F(y, u, v, \lambda, \alpha) = F_1(y, u, v, \lambda, \alpha) - F_2(y, u, v, \lambda, \alpha),$$

where F_1 and F_2 are the following convex functions

$$F_1(y, u, v, \lambda, \alpha) = f_1(\lambda - \alpha, y), \qquad F_2(y, u, v, \lambda, \alpha) = f_2(\lambda - \alpha, y).$$

DCA for (12): By assumption, the feasible set C of (12) is a bounded polyhedral convex set. Its indicator function χ_C is defined by $\chi_C(y,u,v,\lambda,\alpha) := 0$ if $(y,u,v,\lambda,\alpha) \in C$, $+\infty$ otherwise.

With the concavity of the function q, (12) can be rewritten as the following DC program

$$\begin{cases} \min G(y,u,v,\lambda,\alpha) - H(y,u,v,\lambda,\alpha) \\ s.t. \qquad (y,u,v,\lambda,\alpha) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n, \end{cases} \tag{14}$$

with G, H being convex functions defined by $(t > t_1)$

$$G := F_1 + \chi_C, \qquad H := F_2 - tq.$$

Recall that if the function f is convex, then $f_2 = 0$. In this case , H is a polyhedral convex function and (14) is a polyhedral DC program.

According to subsection 3.1, performing DCA for problem (14) amounts to computing the two sequences $\left\{(y^k,u^k,v^k,\lambda^k,\alpha^k)\right\}$ and $\left\{(Y^k,U^k,V^k,\Lambda^k,A^k)\right\}$ defined by

$$(Y^k,U^k,V^k,\Lambda^k,A^k) \in \partial H(y^k,u^k,v^k,\lambda^k,\alpha^k), \tag{15}$$

$$(y^{k+1},u^{k+1},v^{k+1},\lambda^{k+1},\alpha^{k+1}) \in \partial G^*(Y^k,U^k,V^k,\Lambda^k,A^k). \tag{16}$$

In other words, we have to compute the subdifferentials ∂H and ∂G^*. Here we have

$$\partial H(y^k,u^k,v^k,\lambda^k,\alpha^k) = \partial F_2(y^k,u^k,v^k,\lambda^k,\alpha^k) + t\partial(-q)(y^k,u^k,v^k,\lambda^k,\alpha^k), \tag{17}$$

with the explicit computation of $\partial(-q)$ as follows

$$\partial(-q)(y,u,v,\lambda,\alpha) = \sum_{l=1}^{n} \left[\partial(\max(-\lambda_l, -1+u_l-v_l)) + \partial(\max(-\alpha_l, -1-u_l+v_l))\right],$$

with

$$\partial(\max(-\lambda_l,-1+u_l-v_l)) = \begin{cases} (0,0,0,-e^l,0) \text{ if } -\lambda_l > -1+u_l-v_l, \\ (0,e^l,-e^l,0,0) \text{ if } -\lambda_l < -1+u_l-v_l, \\ \left[(0,0,0,-e^l,0),(0,e^l,-e^l,0,0)\right] \\ \text{if } -\lambda_l = -1+u_l-v_l, \end{cases} \tag{18}$$

and

$$\partial(\max(-\alpha_l,-1-u_l+v_l)) = \begin{cases} (0,0,0,0,-e^l) \text{ if } -\alpha_l > -1-u_l+v_l, \\ (0,-e^l,e^l,0,0) \text{ if } -\alpha_l < -1-u_l+v_l, \\ \left[(0,0,0,0,-e^l),(0,-e^l,e^l,0,0)\right] \\ \text{if } -\alpha_l = -1-u_l+v_l, \end{cases} \tag{19}$$

where $e^1,...,e^n$ are the unit vectors of $\mathbb{R}^n$.

As for computing $\partial G^*(Y^k,U^k,V^k,\Lambda^k,A^k)$, we have to solve the following related convex program :

$$\begin{cases} \min F_1(y,u,v,\lambda,\alpha) - \langle (y,u,v,\lambda,\alpha),(Y^k,U^k,V^k,\Lambda^k,A^k)\rangle \\ s.t \qquad (y,u,v,\lambda,\alpha) \in C. \end{cases} \tag{20}$$

We decompose this convex program into two convex programs:

$$\begin{cases} \min_{(y,\lambda,\alpha)} f_1(\lambda-\alpha,y) - \langle (y,\lambda,\alpha), (Y^k, \Lambda^k, A^k)\rangle \\ s.t. \qquad (\lambda-\alpha,y) \in K, \lambda,\alpha \in [0, M+2]^n, \end{cases} \tag{21}$$

and

$$\begin{cases} \min_{(u,v)} -\langle (u,v), (U^k, V^k)\rangle \\ s.t. \qquad \langle e_n, u+v\rangle \le k, u, v \in [0,1]^n. \end{cases} \tag{22}$$

DCA for (13): For (13) we take

$$G := F_1 + \rho\langle e_n, u+v\rangle + \chi_C, \qquad H := F_2 - tq.$$

As for computing $\partial G^*(Y^k, U^k, V^k, \Lambda^k, A^k)$, we have to solve the following related convex programs :

$$\begin{cases} \min_{(y,\lambda,\alpha)} f_1(\lambda-\alpha,y) - \langle (y,\lambda,\alpha), (Y^k, \Lambda^k, A^k)\rangle \\ s.t. \qquad (\lambda-\alpha,y) \in K, \lambda,\alpha \in [0, M+2]^n, \end{cases} \tag{23}$$

and

$$\begin{cases} \min_{(u,v)} \langle (u,v), (\rho e_n - U^k, \rho e_n - V^k)\rangle \\ s.t. \qquad u, v \in [0,1]^n. \end{cases} \tag{24}$$

Now we can describe DCA applied to (14).

Initialization Let ϵ be the tolerance sufficiently small, set $k = 0$.
Choose $(y^0, u^0, v^0, \lambda^0, \alpha^0) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$.
Repeat

- Compute $(Y^k, U^k, V^k, \Lambda^k, A^k)$ via (17), (18), (19).
- Solve (22) (resp. (24)) to obtain (u^{k+1}, v^{k+1}) and (21) to obtain $(y^{k+1}, \lambda^{k+1}, \alpha^{k+1})$.
- $k+1 \leftarrow k$.

Until
$\|y^{k+1} - y^k\| + \|u^{k+1} - u^k\| + \|v^{k+1} - v^k\|$
$+ \|\lambda^{k+1} - \lambda^k\| + \|\alpha^{k+1} - \alpha^k\| \le \epsilon$.

4 Applications

In this section we present some feature selection problems. Feature selection is an important combinatorial optimization problem in the context of supervised pattern classification. The main goal in feature selection is to select a subset of features of a given data set while preserving or improving the discriminative ability of a classifier.

It is well known [10] that the problem of minimizing the l_0 norm is NP-Hard. Bradley and Mangasarian [11] proposed an approximation method in which the l_0 norm is approximated by a concave exponential function. Weston et al. [1] have used another approximated function.

To test our method we consider the following feature selection problems.

1. Combined feature selection (L2-L0 SVM)

$$(CFS) \qquad \begin{cases} \min_{(w,b,\xi)} \frac{\mu}{m} e_m^T \xi + \frac{1}{2} w^T w + \nu \|w\|_0 \\ s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, ..., m, \\ \quad \xi \geq 0, w \in \mathbb{R}^n, b \in \mathbb{R}. \end{cases}$$

2. Zero-norm for feature selection

$$(ZFS) \qquad \begin{cases} \min_{(w,b,\xi)} \frac{\mu}{m} e_m^T \xi + \frac{1}{2} w^T w \\ s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, ..., m, \\ \quad \|w\|_0 \leq k, \xi \geq 0, w \in \mathbb{R}^n, b \in \mathbb{R}. \end{cases}$$

3. Original FSV

$$(FSV) \qquad \begin{cases} \min_{(w,\gamma,y,z)} f_\lambda(w,\gamma,y,z) = (1-\lambda)(\frac{1}{m} e^T y + \frac{1}{k} e^T z) + \lambda \|w\|_0 \\ s.t. \quad -Aw + e\gamma + e \leq y, Bw - e\gamma + e \leq z, \\ \quad w \in \mathbb{R}^n, y \in \mathbb{R}^m_+, z \in \mathbb{R}^k_+, \gamma \in \mathbb{R}. \end{cases}$$

For the original FSV we have the following analytical results:

- If $a \in \{1...n\}$ and $\lambda \geq \frac{2}{2+a}$, then $\|w\|_0 < a$ or $f_\lambda(w,\gamma,y,z) \geq f_\lambda(0,0,e,e)$ for all feasible point of FSV;
- If $\lambda \geq \frac{2}{3}$ then $(0, \gamma, (1+\gamma)e, (1-\gamma)e)$, $\gamma \in [-1,1]$ are global solutions of FSV;
- If $\lambda \geq 0.05$ then all the global solutions (w,γ,y,z) of FSV satisfy $\|w\|_0 \leq 38$.

This results show the non uniqueness of global solution and that for high number of features it is necessary to choose small values for λ.
Many questions can arise

- A global solution gives better one classification than a local solution?
- Model with the original l_0 norm gives better one classification than model with the l_0 approximations?
- How to choose good parameters?

In the next section we present some preliminary results in the context of SVMs.

5 Experiments and Preliminary Numerical Results

We have implemented our method for (CFS) and (ZFS) and we start DCA with $x^0 = e$ and stop DCA on a solution with tolerance $tol = 10^{-8}$. The number of selected features is determined as $|\{j = 1, ..., n : |w_j| > 10^{-8}\}|$. We take the exact penalty parameter t greater than 1000. For solving quadratic convex problem (21) we use CPLEX 7.5.

5.1 Data Sets

To test our methods on real-world data, we use several data sets from the UCI repository. The problems mostly treat medical diagnoses based on genuine patient data and are resumed in Table 1 where we use distinct short names for databases.

5.2 Numerical Results

In Table 2, we summarize the computational results obtained on (CFS) and (ZFS). We observe from the computational results that

- The classifiers obtained by this method with DCA suppressed many features. The number of features is considerably reduced while the classifier is quite good. For (CFS) the correctness of the classification on the test set vary from 57.90% (when 83.33% of features are deleted) to 97.40% (when 33.33% of features are deleted). For (ZFS) the correctness of the classification on the test set vary from 57.90% (when 83.33% of features are deleted) to 87.30% (when 88.88% of features are deleted).
- DCA realizes the suitable trade-off between the error of classification and the number of features.

Table 1. Statistics for data sets used

data set	no. of features	no. of samples	class dist.(+)/(-)
ionosphere	34	351	225/126
breast cancer wisconsin	9	683	444/239
wdbc wisconsin	30	569	212/357
wpbc wisconsin	32	198	151/47
Bupa Liver	6	345	145/200

Table 2. Computational results

	(CFS)			(ZFS)		
data set	selected features	train correctness (%)	test correctness (%)	selected features	train correctness (%)	test correctness (%)
ionosphere	1	76.10	70.95	1	73.08	78.64
bcw	6	92.80	97.40	1	88.90	87.30
wdbc wisconsin	4	93.13	92.64	2	69.32	69.74
wpbc wisconsin	4	83.10	74.44	3	84.62	74.70
Bupa Liver	1	58.10	57.90	1	58.10	57.90

6 Conclusion

We have proposed in this paper a new DC programming approach for solving problem dealing l_0 norm. The resulted DC program is polyhedral in general and DCA has a finite convergence.

Preliminary computational results show that the proposed approach is promising for feature selection in the context of SVMs. Studies in the large-dimension case and large-dimension data are in progress.

References

1. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the zero−norm with linear models and kernel methods. Journal of Machine Learning Research 3, 1439–1461 (2003)
2. Neumann, J., Schnörr, C., Steidl, G.: Combined SVM-based Feature Selection and classification. Machine Learning 61, 129–150 (2005)
3. Schnörr, C.: Signal and image approximation with level-set constraints. Computing 81, 137–160 (2007)
4. Le Thi, H.A., Pham Dinh, T., Le Dung, M.: Exact Penalty in DC Programming. Vietnam Journal of Mathematics 27(2), 169–179 (1999)
5. Le Thi, H.A., Pham Dinh, T., Huynh Van, N.: Exact Penalty Techniques in DC Programming, Research Report, National Institute for Applied Sciences, Rouen (submitted, 2004)
6. Luo, Z.-Q., Pang, J.-S., Ralph, D., Wu, S.-Q.: Exact Penalization and Stationarity Conditions of Mathematical Programs with Equilibrium Constraints. Mathematical Programming 75, 19–76 (1996)
7. Pham Dinh, T., Le Thi, H.A.: Convex Analysis Approaches to DC Programming: Theory, Algorithms and Applications. Acta Mathematica Vietnamica 22(1), 287–367 (1997)
8. Rockafellar, R.T.: Convex Analysis. Princeton University Press, N.J (1970)
9. Vicente, L.N., Calamai, P.H.: Bilevel and Multilevel Programming: A Bibliography Review. Journal of Global Optimization 5, 291–306 (1994)
10. Amaldi, E., Kann., V.: On the approximability of minimizing non zero variables or unsatisfied relations in linear systems. Theoretical Computer Science 209, 237–260 (1998)
11. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Proceeding of International Conference on Machina Learning ICML 2008 (2008)
12. Thiao, M., Pham Dinh, T., Le Thi, H.A.: Exact reformulations for a class of problems dealing l_0 norm. Technical report, L.M.I INSA of Rouen (2008)

Finding Maximum Common Connected Subgraphs Using Clique Detection or Constraint Satisfaction Algorithms

Philippe Vismara[1,2] and Benoît Valery[1]

[1] LIRMM - 161 rue Ada - 34392 Montpellier Cedex 5 - France
[2] LASB - Montpellier SupAgro - 2 place Pierre Viala - 34060 Montpellier Cedex 1
vismara@lirmm.fr, valery@lirmm.fr

Abstract. This paper investigates the problem of Maximum Common Connected Subgraph (MCCS) which is not necessarily an induced subgraph. This problem has so far been neglected by the literature which is mainly devoted to the MCIS problem. Two reductions of the MCCS problem to a MCCIS problem are explored: a classic method based on linegraphs and an original approach using subdivision graphs. Then we propose a method to solve MCCS that searchs for a maximum clique in a compatibility graph. To compare with backtrack approach we explore the applicability of Constraint Satisfaction framework to the MCCS problem for both reductions.

Keywords: Maximum common subgraph; linegraph; subdivision graph, compatibility graph; constraints satisfaction algorithm; clique detection.

1 Introduction

A classic method for comparing two graphs is to find the largest pattern between them. Most of the time, this question is interpreted as a *maximum common induced subgraph* (MCIS) problem. Nevertheless, some slightly different problems can be relevant in many areas. For instance, finding connected subgraphs can be preferred to compare molecules in the design of organic synthesis.

In this paper we investigate the problem of *Maximum Common Connected Subgraph* (MCCS) which is not necessarily an induced subgraph. This problem has so far been neglected by the literature which is mainly devoted to the MCIS problem.

The algorithms that solve MCIS are generally classified into two main categories: backtrack algorithms and methods that find a maximum clique in a *compatibility graph*. This latter approach is one of the most popular and is generally based on variants of the Bron and Kerbrosch's algorithm [3] that finds the maximal cliques of a graph. Koch [8] has proposed an extension of the method adapted to the MCCIS problem (connected MCIS). The non-clique based backtrack approach is symbolized by McGregor's algorithm [12]. This method has several similarities with the framework of Constraint Satisfaction Problems.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 358–368, 2008.

Common subgraph problems for chemical structures matching are explored in [14]. The MCIS problem is NP-hard except for almost trees of bounded degree [1]. As for the MCCIS problem, it is polynomial for partial k-tree [18].

Based on Withney's theorem [17] on *linegraphs*, a reduction of the MCS problem (neither induced nor connected) to a MCIS problem is often suggested (but never detailed) in the literature. In this paper we explore this reduction for MCCS on labeled graphs. We also investigate another reduction based on the *subdivision graph* notion. Then we study how to solve this problem using a clique-based algorithm for both reductions. In section 4 we explore the applicability of constraint satisfaction algorithms to the MCCS problem. For each approach we compare the efficiency of using linegraphs or subdivision graphs to transform the problem into a MCCIS problem. Experimental results are reported in section 5.

2 Preliminaries

We consider connected graphs with labeled nodes and edges. Formally, a graph is a 4-tuple, $G = (V, E, \mu, \nu)$, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges, $\mu : V \to L_V$ is a function assigning to each vertex a label from the set of labels L_V and similarly $\nu : E \to L_E$ is the edge labeling function.

For any edge $e = xy$ we define $ends(e) = \{x, y\}$.

A graph $H = (V', E', \mu', \nu')$ is a *subgraph* of G iff $V' \subseteq V$, μ' and ν' are the restrictions of μ and ν respectively and $E' \subseteq E \cap (V' \times V')$. The graph H is an *induced subgraph* of G if $E' = E \cap (V' \times V')$.

Given two graphs G_1 and G_2, a Common Connected Subgraphs (CCS) of G_1 and G_2 is a connected graph H isomorphic to both subgraphs of G_1 and G_2.

A Maximum Common Connected Subgraphs (MCCS) is a Common Connected Subgraphs which size is maximum according to the number of edges. By analogy, we can define a Maximum Common Connected Induced Subgraph (MCCIS). Generally, MCCIS is maximum according to number of vertices. Figure 1 illustrate differences between MCS, MCCS, MCIS, and MCCIS.

Linegraph. The linegraph $L(G)$ of a graph $G = (V, E, \mu, \nu)$ is a graph that has a vertex for each edge of G, and two vertices of L(G) are adjacent if they correspond to two edges of G with a common extremity.

According to Withney's theorem [17], two connected graphs with isomorphic linegraphs are isomorphic unless one is a triangle (K_3) and the other is a trinode

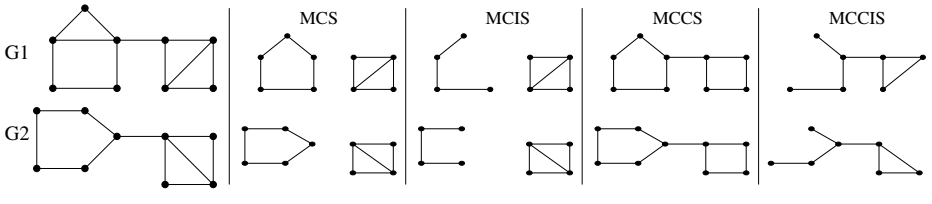


Fig. 1. Differences between MCS, MCCS, MCIS, and MCCIS

($K_{1,3}$) since both graphs have their linegraph equal to K_3. Hence, the MCCS problem between two unlabeled connected graphs can be solved as an MCCIS problem between their linegraphs. Checking for triangle / trinode exchange must be done only for solutions including less than 4 edges.

It is important to note that there is no such a direct equivalence for the MCS problem (not necessarily connected) because a MCIS between two linegraphs can have many connected components reduced to K_3. Since solutions with triangle / trinode exchanges can be larger than solutions without exchanges, the test for exchanges must be done during the search.

Now we consider the MCCS problem for labeled graphs. To insure the equivalence with MCCIS for labeled graphs, the corresponding linegraphs must be labeled on both nodes and edges (see figure 2). We define the labeling functions of the linegraph $L(G) = (E, \mathcal{E}, \mu_L, \nu_L)$ as follows: $\forall e \in E,$ where $e = xy,\ \mu_L(e) = (\nu(e), \mu(x), \mu(y))$ and $\forall \alpha\beta \in \mathcal{E},\ \nu_L(\alpha\beta) = \mu(ends(\alpha) \cap ends(\beta))$. Using this definition, Withney's theorem can be extended to labeled graphs. To demonstrate this result, one can adapt the proof presented in [6]. Given an isomorphism of $L(G_1)$ onto $L(G_2)$ preserving the labels, it is easy to derive an isomorphism of G_1 onto G_2 that preserves the labels.

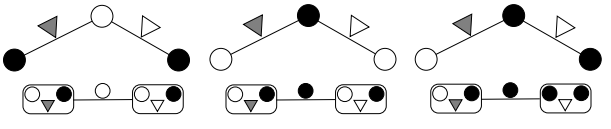


Fig. 2. Labeling linegraphs

Subdivision Graph. The *subdivision graph* $S(G)$ is obtained from a graph $G = (V, E, \mu, \nu)$ by replacing each edge $e = xy$ by a new vertex e connected to both x and y.

Formally, $S(G) = (V \cup E, \mathcal{E}, \mu_s, f_0)$ where $\mathcal{E} = \{x\alpha \in V \times E \mid x \in ends(\alpha)\}$, f_0 is the zero function and $\forall x \in V, \mu_s(x) = \mu(x)$ and $\forall e \in E, \mu_s(e) = \nu(e)$.

Definition 1. *A balanced subgraph of a subdivision graph* $S(G) = (V \cup E, \mathcal{E}, \mu_s, f_0)$ *is a subgraph in which any vertex from* E *has an odd degree.*

Then, the MCCS problem between two graphs is clearly equivalent to find maximum common connected and balanced subgraphs between their subdivision graphs.

3 Clique Detection

The detection of a MCIS between two graphs (G_a and G_b) can be solved by finding maximum clique in the *compatibility graph* (G_C). A *compatibility graph* of two graphs, also called modular graph, is a graph whose node set is $V_a \times V_b$. A node (x_i, x_j) in G_C represents a mapping between the vertex x_i from G_a and the vertex x_j from G_b. An edge between two nodes in G_C represents two compatible

mapping. Then a clique in G_C of size k is a compatible mapping of k vertices in G_a with k vertices of G_b.

Reducing the MCIS problem to the maximum clique has been discovered independently by numerous authors such as Levi [11]. Clique detection is a common approach to compute MCIS. I. Koch [8] proposed a method to find MCCIS involving labels on edges of the compatibility graph.

In this section we present the ways to solve MCCS using clique detection algorithms on compatibility graphs constructed from linegraphs or subdivision graphs.

Clique Detection Based on Linegraphs

According to section 2, we can reduce MCCS to MCCIS. The compatibility graph is constructed with $L(G_1)$ and $L(G_2)$ instead of G_1 and G_2.

Given two labeled linegraphs $L(G_1) = (E_1, \mathcal{E}_1, \mu_1, \nu_1)$ and $L(G_2) = (E_2, \mathcal{E}_2, \mu_2, \nu_2)$, the compatibility graph $G_C = (V_C, E_C, f_0, \nu_C)$ is defined as :

- $V_C = \{(x_1, x_a) \in E_1 \times E_2 \mid \mu_1(x_1) = \mu_2(x_a)\}$
- $E_C = \{(x_1, x_a)(x_2, x_b) \in V_C \times V_C\}$ such that :
  - $x_1 \neq x_2$ and $x_a \neq x_b$ and
  - $(x_1x_2 \in \mathcal{E}_1$ and $x_ax_b \in \mathcal{E}_2)$ or $(x_1x_2 \notin \mathcal{E}_1$ and $x_ax_b \notin \mathcal{E}_2)$
- $\nu_C : E_C \rightarrow \{\text{strong}, \text{weak}\}$ such that :
  - $\nu_C((x_1, x_a)(x_2, x_b)) = \begin{cases} \text{strong if } (x_1x_2 \in \mathcal{E}_1 \text{ and } x_ax_b \in \mathcal{E}_2) \\ \text{weak otherwise} \end{cases}$

A *clique* is a subset of nodes such that each pair of nodes is connected by an edge. An edge e is a *strong* edge iff $\nu_C(e) = \text{strong}$. Two nodes a and b in G_C are said *strongly* (resp. *weakly*) *connected* iff ab is a strong edge (resp. weak). A clique is a *strong clique* if it contains a covering tree that consists of strong edges. Hence, the common subgraph corresponding to a strong clique is necessarily connected.

Once the compatibility graph is constructed (in $O(|E_1| \times |E_2|)$), a clique detection algorithm is used to find maximum cliques. The maximum clique problem is a classical problem in combinatorial optimization and has been widely studied [2]. The Bron and Kerbrosch's algorithm[3] is one of the first and most popular. This algorithm computes all *maximal cliques* but is often used for finding the *maximum clique*. The benefit of the backtracking method in [3] is that it avoids generating non-maximal cliques.

This algorithm has known several modifications such as Johnston's heuristic [7]. During the backtrack search, once the current clique K has been extended with z, K must be extended without z. [7] showed that the next node y to extend K can be taken within nodes disconnected to z since any maximal clique without z must include such a node.

Koch's algorithm [8] is based on [3] and computes all maximal strong cliques. The current clique K is extended with a node z strongly connected to K. Unfortunately, Johnston's heuristic[7] cannot be applied since nodes disconnected to z aren't necessarily strongly connected to K (see fig. 3). We propose to modify the heuristic such that the next node y (strongly connected to K) is added to K if either y is disconnected to z or y is strongly connected to a node t weakly

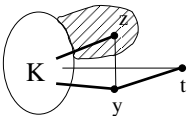


Fig. 3. Extending the current strong clique

connected to K and such that t is disconnected to z. In this way, the maximum strong clique found represents a MCCIS of $L(G_1)$ and $L(G_2)$ and therefore a MCCS of G_1 and G_2.

Clique Detection Based on Subdivision Graphs
The compatibility graph can be created upon $S(G_1)$ and $S(G_2)$. As far as we know, subdivision graphs have not been used to reduce the MC(C)S problems to the MC(C)IS problems. Since a subdivision graph has two different kinds of nodes, the construction of the compatibility graph is more tricky.

The construction of the compatibility graph must ensure that a maximal clique corresponds to a balanced subgraph in $S(G_1)$ and $S(G_2)$.

Given two subdivision graphs $S(G_1) = (V_1 \cup E_1, \mathcal{E}_1, \mu_1, f_0)$ and $S(G_2) = (V_2 \cup E_2, \mathcal{E}_2, \mu_2, f_0)$, the compatibility graph $G_C = (V_C \cup E_C, \mathcal{E}_C, f_0, \nu_C)$ is defined as follows:

- $V_C = \{(x_1, x_a) \in V_1 \times V_2 \mid \mu_1(x_1) = \mu_2(x_a)\}$
  $E_C = \{(e_1, e_a) \in E_1 \times E_2 \mid \nu_1(e_1) = \nu_2(e_a)$ and $\mu_1(ends(e_1)) = \mu_2(ends(e_2))\}$
- $\mathcal{E}_C =$
  1. $\{(x_1, x_a)(e_1, e_a) \in V_C \times E_C \mid (x_1 \in ends(e_1)$ and $x_a \in ends(e_a))$ or $(x_1 \notin ends(e_1)$ and $x_a \notin ends(e_a))\} \cup$
  2. $\{(x_1, x_a)(x_2, x_b) \in V_C \times V_C \mid x_1 \neq x_2$ and $x_a \neq x_b\} \cup$
  3. $\{(e_1, e_a)(e_2, e_b) \in E_C \times E_C\}$ such that :
     - $e_1 \neq e_2$ and $e_a \neq e_b$ and
     - $|ends(e_1) \cap ends(e_2)| = |ends(e_b) \cap ends(e_b)|$
- $\nu_C : \mathcal{E}_C \rightarrow \{\text{strong}, \text{weak}\}$ such that :
  - $\nu_C((x_1, x_a)(e_1, e_a)) = \begin{cases} \text{strong if } x_1 \in ends(e_1) \text{ and } x_a \in ends(e_a) \\ \text{weak otherwise} \end{cases}$
  - $\nu_C((x_1, x_a)(x_2, x_b)) = \nu_C((e_1, e_a)(e_2, e_b)) = \text{weak}$

The time complexity of the construction is $O((|V_1| + |E_1|) \times (|V_2| + |E_2|))$.

Theorem 1. *A maximal strong clique K of the compatibility graph G_C defines a balanced connected subgraph in the subdivision graphs $S(G_1)$ and $S(G_2)$.*

Proof. The main result is to prove that for any $(a, b) \in E_C$, if $(a, b) \in K$ then K must include a couple of nodes in V_C that maps the ends of a and b. Let $(a, b) \in E_C \cap K$ such that $ends(a) = \{x, y\}$ and $ends(b) = \{i, j\}$. By construction (1) of $\mathcal{E}_C$, (a, b) is strongly connected to (x, i), (x, j), (y, j) and (y, i). By (2), (x, i) is adjacent to (y, j) and (x, j) is adjacent to (y, i). For any node $(z, k) \in V_C \cap K$ where $z \notin ends(a)$ or $k \notin ends(b)$, since (z, k) is adjacent to (a, b) we have by (1) (z, k) is adjacent to (x, i), (x, j), (y, j) and (y, i). Now let's

assume that K includes another node $(c,d) \in E_C$. By (3), let $\alpha = |ends(a) \cap ends(c)| = |ends(b) \cap ends(d)|$. If $\alpha = 0$, then (c,d) is necessarily adjacent to (x,i), (x,j), (y,j) and (y,i). If $\alpha = 1$, without loss of generality, suppose that $ends(a) \cap ends(c) = \{x\}$ and $ends(b) \cap ends(d) = \{i\}$. Thus (c,d) is strongly connected to (x,i). Since $y \notin ends(c)$ and $j \notin ends(d)$ the nodes (y,j) and (c,d) must be connected. Hence, K must include (x,i) and (y,j) to be maximal.

4 Constraint Satisfaction Problems

In [12], McGregor presents one of the rare algorithms specially intended for MCS problems. Even so, this method has often been used to solve MCIS problems. The method is based on a backtrack algorithm but the way the method is implemented has some analogy with the general framework of Constraint Satisfaction Problems (CSP). Constraint satisfaction algorithms have been applied to several problems in Graph Theory [10,5] but the MCCS problem has not yet been formulated as a constraint satisfaction problem. Since CSP research finding can benefit such an approach, we chose to study the applicability of backtrack methods to MCCS in the scope of CSP.

From Induced Subgraph Problem to MCCIS. A constraint satisfaction problem (CSP) is described by a constraint network defined as a triple whose elements are a set of variables $X = \{x_1, x_2, ..., x_k\}$, a set of values for each variable, and a set of constraints among variables to specify which tuples of values can be assigned to tuples of variables. A solution of the CSP is an instantiation $\mathcal{I}$ of the variables that satisfies all the constraints.

For instance, a CSP for checking whether a graph G_1 is an induced subgraph of a graph G_2 could be defined as follows: (i) a variable X_i is defined for each vertex i of G_1; (ii) a variable X_i can be assigned to any vertex of G_2 whose label is the same as that of i; the set of values that X_i can take is called the domain of X_i and denoted by $D(X_i)$; (iii) a binary constraint $C(X_i, X_j)$ is defined between each pair of variables X_i, X_j to insure that the connectivity and the labeling are preserved by the mapping. A pair of values $(y_i, y_j) \in D(X_i) \times D(X_j)$ is allowed by the constraint if $ij \in E_1 \Leftrightarrow y_i y_j \in E_2$ and when $ij \in E_1, \nu_1(ij) = \nu_2(y_i y_j)$; (iv) a constraint of difference [15] is defined on variables to ensure that they all take different values. Any solution for this constraint network is a matching of G_1 to G_2.

Although the standard approach to solve a CSP is based on backtracking, the reader interested in algorithms to solve CSP should refer to the vast literature on this domain [16]. In this paper we focus on the classical constraint network framework which is oriented towards the satisfaction of all constraints. This framework is widely used and has been implemented in many constraint programming toolkits as JChoco [9] a Java library for constraint programming. In the last decade, several extensions of the classical CSP framework have been proposed. Some of them, like *soft constraints*, could be interesting to solve the MCCS problem. But most of the CSP solvers do not implement these extensions. Hence, they have not been studied yet in the context of the present work.

In the previous example we have defined a constraint network to solve the induced subgraph isomorphism problem. Representing an MCIS problem should differ in the way that some vertices of graph G_1 are not mapped to any vertex of G_2. Hence, the corresponding variables of the CSP cannot be assigned to values in X_2. A usual solution in such a case is to add an extra value (we denote $\star$) to the domain of the variables. Note that the constraint of difference must be weakened since many variables can be assigned to the $\star$ value. Then, for any solution of the CSP, the common induced subgraph will correspond to the variables assigned to values in X_2 only. The size of the common induced subgraph is the number of variables whose value differs from $\star$. Solving the MCIS problem is then equivalent to find a solution of the constraint network that minimize the number of $\star$ values.

To solve the MCCIS problem we add a new global constraint to the previous CSP. This *connectivity constraint* checks the connectivity of all the vertices whose corresponding variable is not assigned to $\star$.

In the following sections we detail the constraint networks to solve the MCCS problem using linegraph or subdivision graph respectively.

A Constraint Network Based on Linegraphs

Given two linegraphs $L(G_1) = (E_1, \mathcal{E}_1, \mu_1, \nu_1)$ and $L(G_2) = (E_2, \mathcal{E}_2, \mu_2, \nu_2)$, we propose to define a network constraint as follows:

- a set of variables $X = \{X_i \mid i \in E_1\}$
- a domain for each variable: $\forall i \in E_1$, $D(X_i) = \{y \in E_2 \mid \mu_2(y) = \mu_1(i)\} \cup \{\star\}$
- a binary constraint $C(X_i, X_j)$ between each pair of variables that allows the set of couples: $\{(k,l) \mid k,l \in E_2 \text{ and } k \neq l \text{ and } (k,l) \in \mathcal{E}_2 \Leftrightarrow (i,j) \in \mathcal{E}_1\}$
  $\cup \{(t,\star),(\star,t) \mid t \in E_2\} \cup \{(\star,\star)\}$
- a global *constraint of connectivity* on X to insure that the subgraph induced by $\{i \in E_1 \mid \mathcal{I}(X_i) \neq \star\}$ is connected, where $\mathcal{I}$ is an instantiation of the variables.

The main difficulty lies in the implementation of the *constraint of connectivity*. We maintain two sets during the search. *CComp* is the set of variables already instancied to a non-$\star$ value and such that the subgraph $\{i \in E_1 \mid X_i \in CComp\}$ is connected. The set *Candidates* includes uninstanciated variables connected to at least one variable in *CComp*. Only variables in *Candidates* can be assigned to non-$\star$ values. The sets are updated after each assignment.

Finally, to minimize the number of $\star$ values a new variable $X_{\#\star}$ is usually added to the constraint network that counts the number of variables assigned to this value. Hence, $D(X_{\#\star}) = \{0 \ldots |E_1|\}$ and a global constraint is defined on $\{X_i\}_{i \in E_1} \cup \{X_{\#\star}\}$ to ensure that $\mathcal{I}(X_{\#\star}) = |\{i \in E_1 \mid \mathcal{I}(X_i) = \star\}|$

A Constraint Network Based on Subdivision Graphs

The constraint network based on subdivision graphs is quite similar to that defined in the previous section. Given two graphs $G_1 = (V_1, E_1, \mu_1, \nu_1)$ and $G_2 = (V_2, E_2, \mu_2, \nu_2)$ and their subdivision graphs $S(G_1) = (V_1 \cup E_1, \mathcal{E}_1, \mu_{s1}, f_0)$

and $S(G_2) = (V_2 \cup E_2, \mathcal{E}_2, \mu_{s2}, f_0)$, we propose to define the network constraint as follows:

1. a set of variables $X = \{X_i \mid i \in V_1 \cup E_1\}$
2. $\forall i \in V_1$, $D(X_i) = \{y \in V_2 \mid \mu_2(y) = \mu_1(i)\} \cup \{\star\}$ and $\forall j \in E_1$, $D(X_j) = \{y \in E_2 \mid \nu_2(y) = \nu_1(j)\} \cup \{\star\}$
3. a binary constraint $C(X_i, X_j)$ for each couple $i, j \in V_1 \times E_1$ that allows the set $\{(k,l) \in V_2 \times E_2 \mid (k,l) \in \mathcal{E}_2 \Leftrightarrow (i,j) \in \mathcal{E}_1\} \cup \{(t,\star), \mid t \in V_2\} \cup \{(\star,\star)\}$
4. a binary constraint between each pair of variables in $\{X_i\}_{i \in V_1}$ (resp. $\{X_j\}_{j \in E_1}$) that forbids equals values except of $\star$.
5. a global constraint of connectivity on X to insure that the subgraph induced by $\{i \in E_1 \mid \mathcal{I}(X_i) \neq \star\}$ is connected, where $\mathcal{I}$ is an instanciation of the variables.

The main difference with the CSP based on linegraphs lies in the partition of the set of variables. By the binary constraint (3) between a "vertex variable" and an "edge variable" we can easily prove that $\forall j \in E_1, \mathcal{I}(X_j) \neq \star \Rightarrow \forall i \in ends(j), \mathcal{I}(X_i) \neq \star$

Hence any solution of the CSP corresponds to a balanced subgraph of the subdivision graphs.

5 Experimental Results

We have implemented our constraint networks with the Java constraint programming library JChoco [9]. The set *CComp* and *Candidates* for the connectivity constraint are handled with backtrackable structures provided by JChoco.

The Bron and Kerbrosch's algorithm [3] for clique detection has been implemented in Java. We modified the program with Koch's work [8] to find only connected solutions. Then we adapted Johnston's heuristic [7] for improving performance.

Our database consists of three sets of graphs. The first one consists of 30 undirected connected graphs without label and randomly generated. Each has at least 10 nodes and at most 20 nodes. The second set of graphs contains 30 molecules with a size between 6 and 62 atoms. The last set of graphs are molecules taken from 5487 chemical reactions. In this set, we only compare the reactant graphs with the product graphs without labels in the same reaction. A timeout was set to 10 minutes for each test.

The first statement can be done about the comparison between subdivision and the linegraph method in either CSP or clique-based approach. The subdivision method are almost always slower than the linegraph method. Subdivision graphs increase the number of nodes of a graph. Since the complexity of MCCS depends of the size of the data, the size of subdivision graphs probably slows the procedure.

For small graphs or labbeled graphs, both CSP and clique approachs solve the same number of problems within the time limit. The difference is more important for larger graphs.

Table 1. Description of the database. The third column (resp. fourth) represents the average size of the compatibility graph on linegraphs (resp. subdivisions graphs).

	# of couples of graphs	average $(m_1 * m_2)$	average $(n_1 \times n_2) + (m_1 + \times m_2)$
molecules	465	475	1019
random graphs	465	835	1816
chemical reactions	8437	169	331

Table 2. Percent of solved problems on the different graphs sets within the time limit

	CSP		Clique detection	
	Linegraph	Subdivision	Linegraph	Subdivision
molecules without labels	73,99%	69,04%	65,6%	55,4%
molecules	98.15%	94.45%	99.2%	97.09%
random graphs	73.6%	71.7%	65.5%	48.5%
chemical reactions	99.68%	99,58%	99,79%	98,62%

One explanation could be that the library used to implement the CSP algorithms is quite complex and not very efficient for small problems. Conversely, the clique detection algorithms are easier to implement but they do not benefit of the CSP heuristics for large problems. As long as the compatibility graph has a reasonable size (see column 3 and 4 from Table 1), the maximum clique can be found within the time limit. When the size of the compatibilty graph arises, finding the maximum clique is harder and the algorithms timeout.

Even with a preliminary benchmark and a different problem, we have a similar conclusion than [4] that deals with MCIS on directed graphs: we cannot point clearly a faster method in general. Meanwhile, its seems that the size of the compatibility graph could be a threshold where the clique detection algorithms become less effective than constraints satisfaction algorithms.

6 Conclusion

We have presented two methods to reduce the MCCS problem to an MCCIS problem. The first one is an adaptation of the reduction based on linegraphs for the "induced" versions of the problems. The second method is a new approach that involves subdivision graphs. As far as we know, this reduction have not been yet applied even for the MCIS problem.

These methods have been formalized in the general scope of labeled graphs.

We have adapted Koch's algorithm [8] that computes an MCCIS by searching a clique in a labeled compatibility graph. We proposed an heuristic for the choice of the next vertex to add to the strong clique. We have extended the model of compatibility graph in order to solve the MCCS problem for both linegraph and subdivision graph reductions.

To investigate backtrack algorithms as McGregor's algorithm[12] we have chosen the general framework of Constraint Satisfaction Problems. We have studied the applicability of constraint satisfaction techniques for linegraphs and subdivision graphs reductions. The constraints we propose are quite simple and may be improved. We have implemented both constraint networks using the JChoco[9], a open source constraint programming toolkit, using the Java programming language.

The four methods we have investigated to solve the MCCS have been implemented in Java. We have experimented these algorithms on molecular labeled graphs and unlabeled random graphs. The first results show that a linegraph approach is generally faster than methods using subdivision graphs. One explanation could be that subdivision graphs include more nodes than the corresponding linegraphs. This drawback could be reduced with heuristics that exploit the specific structure of subdivision graphs. For small or very labelled graphs there is no significant difference between CSP and clique approachs. For more complex graphs, it seems that the clique detection algorithms become less effective than constraint algorithms.

Since the constraint networks we have proposed are based on quite simple constraints, it would be interesting to optimize them. Another solution would be to use extensions of the classical constraint network framework as *soft constraints*[13].

Nevertheless, the MCCS problem itself has many variants for real word problems. For instance we can use different criteria to calculate the size of a common subgraph (number of nodes and edges, ...). It should be interesting to compare the different methods according to their adaptability to these variations.

References

1. Akutsu, T.: A polynomial time algorithm for finding a largest common subgraph of almost trees of bounded degree. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E76-A(9) (1993)
2. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Du, D.-Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization (Supplement vol. A), pp. 1–74. Kluwer Academic, Dordrecht (1999)
3. Bron, C., Kerbosch, J.: Finding all cliques of an undirected graph. Communication of the ACM 16(9), 575–579 (1973)
4. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI 18(3), 265–298 (2004)
5. Dooms, G., Deville, Y., Dupont, P.: Cp(graph): Introducing a graph computation domain in constraint programming. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709. Springer, Heidelberg (2005)
6. Harary, F.: Graph Theory. Addison-Wesley, Reading (1969)
7. Johnston, H.C.: Cliques of a graph-variations on the bron-kerbosch algorithm. International Journal of Computer and Information Sciences 5(3), 209–238 (1976)
8. Koch, I.: Enumerating all connected maximal common subgraphs in two graphs. Theoretical Computer Science 250, 1–30 (2001)

9. Laburthe, F., Jussien, N.: Jchoco: A java library for constraint satisfaction problems, http://choco.sourceforge.net
10. Larossa, J., Valiente, G.: Constraint satisfaction algorithms for graph pattern matching. Math. Struct. Comput. Sci. 12(4), 403–422 (2002)
11. Levi, G.: A note on the derivation of maximal common subgraphs of two directed or undirected graphs. Calcolo 9(4), 341–352 (1972)
12. McGregor, J.J.: Backtrack search algorithms and the maximal common subgraph problem. Software Practice and Experience 12, 23–34 (1982)
13. Meseguer, P., Rossi, F., Schiex, T.: Soft constraints. In: Rossi, et al. (eds.) [16], pp. 281–328
14. Raymond, J.W., Willett, P.: Maximum common subgraph isomorphism algorithms for the matching of chemical structures. Journal of Computer-Aided Molecular Design 16(7), 521–533 (2002)
15. Régin, J.-C.: A filtering algorithm for constraints of difference in CSPs. In: AAAI 1994, Proceedings of the National Conference on Artificial Intelligence, Seattle, Washington, pp. 362–367 (1994)
16. Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming. Elsevier, Amsterdam (2006)
17. Whitney, H.: Congruent graphs and the connectivity of graphs. Am. J. Math. 54, 150–168 (1932)
18. Yamaguchi, A., Mamitsuka, H., Aoki, K.F.: Finding the maximum common subgraph of a partial k-tree and a graph with a polynomially bounded number of spanning trees. Information Processing Letters 92(2), 57–63 (2004)

Analysis and Solution Developement of the Single-Vehicle Inventory Routing Problem

Yiqing Zhong and El-Houssaine Aghezzaf

Department of Industrial Management, Ghent University,
Technologiepark 903, B-9052 Zwijnaarde, Belgium
Elhoussaine.aghezzaf@ugent.be, Yiqing.Zhong@ugent.be

Abstract. The inventory routing problem (IRP) is a challenging optimization problem underlying the vendor managed inventory policy. In this paper, we focus on a particular case of this problem, namely, the long-term single-vehicle IRP with stable demand rates. The objective is thus to develop an optimal cyclical distribution plan, of a single product, from a single distribution center to a set of selected customers. After an analysis of the problem's features, we propose and discuss a hybrid approximation algorithm to solve the problem. The approach is then tested on some randomly generated problems to evaluate its performance.

1 Introduction

The inventory routing problem (IRP) is one of the challenging optimization problems in logistics and distribution. The problem is of special interest, to supply chain managers in particular, since it provides them with integrated plans that coordinate inventory control with vehicle scheduling and routing policies. In practice, policies such as 'Vendor Managed Inventory' (VMI) are actually used to coordinate inventory control and delivery scheduling in the supply chain. VMI refers to an agreement between a vendor and his customers according to which customers allow the vendor to decide the size and timing of their deliveries [1]. In other words, the vendor is granted full authority to manage his customers' inventories. Compared with the traditional nonintegrated replenishments and routings, in which customers manage their inventories themselves and call in their orders, overall inventory and routing performances throughout the supply chain is by far superior when VMI is implemented. The IRP is actually an underlying optimization model for the VMI policy.

In this paper we discuss the particular long-term single-vehicle IRP which is formulated as a nonlinear mixed integer problem. The original model for this case is proposed and discussed in [2]. The model provides optimal solutions in which a single distribution center is supplying a single product to a selected set of customers in a cyclical way. The main objective of this paper is to develop and suggest an exact solution strategy for the single-vehicle sub-problem.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 369–378, 2008.

2 Formulation of Single-Vehicle IRP

The single-vehicle inventory routing problem consists of a single distribution center r distributing a single product to a selected subset of customers from an already located set of customers S. It is assumed that customer demand rates are stable over time and that a single vehicle is available for the distribution of the product. The objective is to select an optimal subset of customers $C \subset S$ to be served by the vehicle and to develop a cyclical distribution strategy that minimizes expected distribution and inventory costs, without causing any stock-outs at any customers during the planning horizon.

2.1 Review of Some Important Concepts

In order to provide a complete description of the problem, we review the concepts of vehicle 'cycle time' and 'multi-tours'. For more detailed description of these concepts see [2].

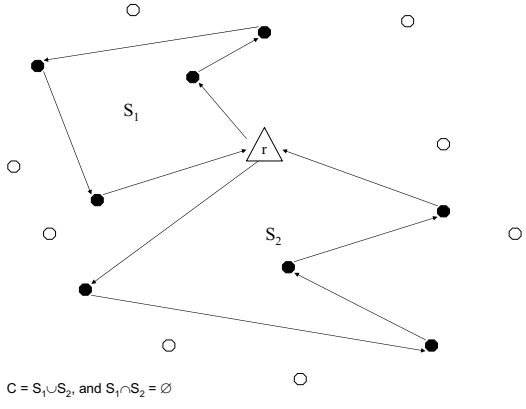


Fig. 1. S is the set of all customers (all nodes), C is the set of selected customers (filled nodes) and the vehicle's trip is made out of two tours, going through the disjoint subsets S_1 and S_2

Consider a vehicle replenishing a set of customers C. Assume that the vehicle makes a trip visiting these customers by means of a set of disjoint tours P. Assume also that each tour $p \in P$ goes through a subset $S_p \cup \{r\}$ of customers such that $C = \bigcup_{p \in P} S_p$ and $S_p \cap S_q = \emptyset$ for any p and $q \neq p$ in P. Under this pattern, the most effective way to supply customers in C is to travel along the traveling salesman tour in each subset $S_p \cup \{r\}$. Let $T_{TSP}(S_p \cup \{r\})$ denote the travel time of each TSP tour on $S_p \cup \{r\}$. Now, if we define the time between two consecutive iterations of the trip as the 'cycle time' and we denote it by $T(C)$. Clearly, this cycle time is bounded from below by the sum of the TSP-tours'

travel times through the subsets $S_p \cup \{r\}$ for $p \in P$. This lower bound is called the 'minimal cycle time' and is also denoted by $T_{min}(C)$. It is given by:

$$T_{min}(C) = \sum_{p \in P} T_{TSP}(S_p \cup \{r\}) \tag{1}$$

There is also an upper bound on the cycle time $T(C)$. It is called 'maximal cycle time' and is denoted by $T_{max}(C)$. This upper bound results from the limited capacity of the vehicle, and is given by:

$$T_{max}(C) = min_{p \in P} \left\{ \frac{\kappa}{\sum_{j \in S_p} d_j} \right\} \tag{2}$$

where d_j is the demand rate of customer $j \in S_p$ and κ is the capacity of the vehicle. The vehicle makes multi-tours S_p and in each tour it cannot carry more than its capacity. This means that $T(C) \leq \kappa / \sum_{j \in S_p} d_j$ for each tour. Therefore $T(C)$ cannot exceed the value of $T_{max}(C)$ given by above. Of course, for a multi-tour going through customers in $C = \bigcup_{p \in P} S_p$ to be feasible, it is necessary that $T_{min}(C)$ be smaller or equal to $T_{max}(C)$.

Finally, there is a theoretical optimal cycle time which can be determined as explained in [2]. This is an extension of the EOQ-formula and is denoted by the 'EOQ cycle time' $T_{EOQ}(C)$. This last value may turn out to be greater than the maximal cycle time or smaller than the minimal cycle time. In these cases, the actual optimal cycle time $T^*(C)$ is equal to the maximal cycle time or minimal cycle time respectively. For a more detailed discussion of these cycle time we refer the reader to in [2], [3] or [4].

2.2 The Modified Formulation for Single-Vehicle IRP

In [2], an approximation algorithm based on column generation is employed to solve the complete nonlinear cyclic long-term IRP. The resulting sub-problems are solved heuristically. In this paper, our discussion will focus mainly on the mixed integer nonlinear formulation of these sub-problems. In the formulation we are presenting in this paper, the nonlinearities in the constraints of the original sub-problem given in [2] are removed. The reformulation of the problem as well as the necessary parameters, variables and constraints are given below.

Additional parameters of the model:

- t_{ij}: The travel time from customer $i \in S^+ = S \cup \{r\}$ to customer $j \in S^+$ (in hours);
- ψ: The fixed operating cost of vehicle (in euro per vehicle);
- δ: The travel cost of vehicle (in euro per km);
- ν: The vehicle speed (in km per hour);
- φ_j: The cost per delivery at customer j (in euro per cycle);
- η_j: The holding cost at customer j (in euro per ton per hour);

– λ_j: The reward (dual price) if customer j is visited. (these prices are obtained from the master problem);

Variables of the model:

– x_{ij}: A binary variable set to 1 if customer $j \in S^+$ is served immediately after customer $i \in S^+$ by the vehicle, and 0 otherwise;
– Q_{ij}: The quantity of the product remaining in the vehicle when it travels directly to the customer $j \in S^+$ from customer $i \in S^+$. This quantity equals zero when the link (i, j) is not on any tour made by the vehicle;
– q_j: The quantity that is delivered to the customer $j \in S$;
– T: The cycle time of the trip made by the vehicle (in hours).

The modified formulation of single-vehicle IRP is given as following:

The Single-Vehicle IRP P1

Minimize

$$RC = \psi + \sum_{i \in S^+} \sum_{j \in S^+} \left((\delta \nu t_{ij} + \varphi_j) \frac{1}{T} + \frac{1}{2} \eta_j d_j T \right) x_{ij} - \sum_{i \in S^+} \sum_{j \in S^+} \lambda_j x_{ij} \quad (3)$$

Subject to:

$$\sum_{i \in S^+} x_{ij} \leq 1, \text{ for all } j \in S, \quad (4)$$

$$\sum_{i \in S^+} x_{ij} - \sum_{k \in S^+} x_{jk} = 0, \text{ for all } j \in S^+, \quad (5)$$

$$\sum_{i \in S^+} \sum_{j \in S^+} t_{ij} x_{ij} - T \leq 0, \quad (6)$$

$$Q_{ij} \leq \kappa \cdot x_{ij} \text{ for all } i, j \in S, \quad (7)$$

$$\sum_{i \in S^+} Q_{ij} - \sum_{k \in S^+} Q_{jk} = q_j, \text{ for all } i \in S^+, j \in S, \quad (8)$$

$$d_j \cdot T \leq q_j + U_j \cdot \left(1 - \sum_{i \in S^+} x_{ij} \right), \text{ for all } j \in S, \quad (9)$$

$$x_{ij} \in \{0, 1\}, Q_{ij} \geq 0, q_j \geq 0, T \geq 0, \text{ for all } i, j \in S^+$$

Constraints (4) make sure that each customer is served at most once. Constraints (5) are the usual flow conservation constraints, insuring that a vehicle assigned to serve a customer will actually serve this customer in one of its tours and will leave to a next customer. Constraints (6) indicate that the cycle time of a vehicle should be greater than the minimal cycle time. Constraints (7) guarantee that the quantity carried by a vehicle doesn't exceed the vehicle's maximum capacity. Constraints (8) are the delivered load balance constraints. In constraints

(9), U_j is given by $U_j = d_j \cdot T_L$, where T_L is the largest possible cycle time. These constraints indicate that the quantity delivered to customer j in one cycle time should be greater than its demand during that time. Note that in $P1$, all constraints are linear.

It is assumed that the time necessary for loading and unloading a vehicle is relatively small in comparison to travel time, and is therefore neglected in this model. Also, inventory capacities at the customers are assumed to be large enough, so corresponding capacity constraints are omitted in the model. Finally, transportation costs are assumed to be proportional to travel times.

3 Analysis and Solution Strategy

3.1 Problem Features

The reformulated model $P1$ is still nonlinear because of the objective function (3). However, for fixed values of the cycle time T, the model becomes linear but mixed integer, and can be solved using an available MILP package. In the first phase of the analysis, we carry out some experiments designed to reveal the structure of the problem's objective function. We solved the problem for a series of fixed values of T given below:

$$T_n^- = \frac{\kappa}{\sum_{i=1}^{n} d_{\sigma(i)}} \text{ and } T_n^+ = \frac{\kappa}{D - \sum_{i=2}^{n} d_{\sigma(i)}}, \tag{10}$$

where n varies from 1 to $|S|$, D is the total demand rate of all customers, and $\left(d_{\sigma(i)}\right)_{(i \in S)}$ is the series of customer demand rates renumbered in the ascending order. T_n^- and T_n^+ are merged in an ascending sequence $\left\{T_{(k)}\right\}$. This is a series of possible values of T covering the feasible domain. It is not sure the optimal T belongs to this series, but this series of cycle times provides some information on the shape of the objective function.

The problem $P1$ is solved for each fixed $T_{(k)}$, in the collection given above, to determine the corresponding optimal solution $X_{(k)}$. For this solution $X_{(k)}$ the minimal, maximal and practical optimal cycle times are obtained and the optimal cycle time among these three is selected. For the selected cycle time the corresponding solution is recomputed again.

Figure 2 depicts some observed features of the problem. The most important is the fact that the objective function is usually non-convex, non-smooth, with many local minima. Also the optimal solution may lie in a very small interval, which imposes some ingenuity is determining the right search step for solution procedure. Examples of these features are graphically shown in Fig. 2. As one may observe in Fig.2, A, B, C, and D the problem is very complex. Notice finally that in cases of B and C, X_1, X_2 and X_3 are different solutions, in which the vehicle visits different sets of customers. Also note that there might exist intervals of T in which there doesn't exist any feasible solution (see Fig. 2.D). Such situations make the development of a solution procedure too difficult.

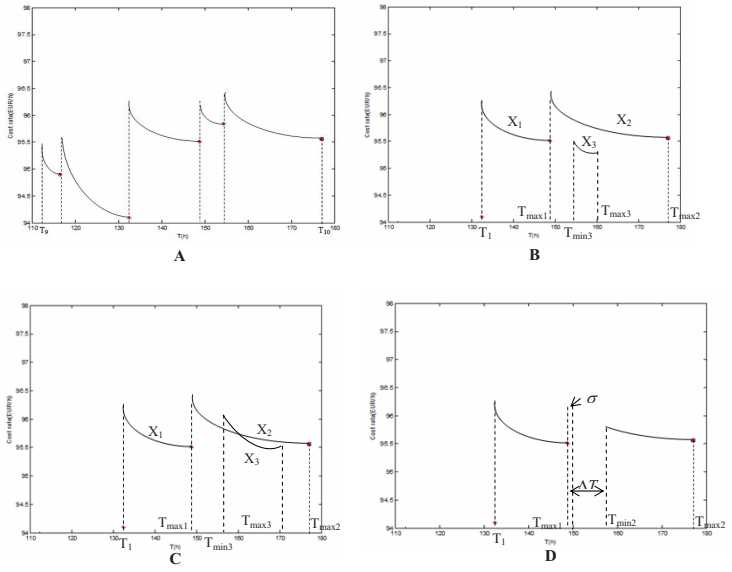


Fig. 2. The example graphs for objective values with T

3.2 Solution Strategy

As shown in the above analysis, the single-vehicle IRP is inherently complex. For its solution, a steepest descent like solution strategy is proposed. Clearly, this process will lead only to an approximate solution. The basic idea is to renew the cycle time step by step and solve the resulting MILP problem for each new cycle time. Recall that S is the set of all customers, the domain for the cycle time variable is $[T_s, T_l]$, where T_s is the smallest value, given by $T_s = \kappa / \sum_{j \in S} d_j$ and T_l is the largest value, given by $T_l = \kappa / \text{Min}_{(j \in S)} \{d_j\}$. Within this domain, a start point $T_k\,(k = 0)$ (*i.e.* T_s) is selected and the problem $P1$ is solved to determine the corresponding optimal solution X_k. Let $T_{min}^{X_k}$ and $T_{max}^{X_k}$ be the solution's minimal and maximal cycle times respectively. The progress is performed by increasing the cycle time as follows:

$$T_{k+1} = T_{max}^{X_k} + \epsilon_k \tag{11}$$

where ϵ_k is a step which could either be constant or variable.

In our discussion, ϵ_k is a small constant selected so as to avoid the possibility of missing any possible optimal solution. It is determined by:

$$\epsilon_k = \left| \frac{\kappa}{d_{i*}} - \frac{\kappa}{d_{j*}} \right| \tag{12}$$

where the customers $i* \in S$ and $j* \in S$ are the ones which give the smallest difference of demand rates (*i.e.* $\Delta d = \text{Min}_{ij} \{|d_i - d_j|\}$). The problem with this approach is that T_{k+1} can be either feasible or infeasible. These two cases are treated respectively as follows:

Case 1. Assume that T_{k+1} is feasible, this means that the steepest descent procedure will work correctly in the interval $\left[T_{k+1}, T_{max}^{X_{k+1}}\right]$. However, we have to be attentive to cases such as B and C shown in Fig. 2. In such cases, by simply increasing $T_{max1} + \epsilon_1$ (assuming T_{max1} is also the practical optimal cycle time for solution X_1), where $\epsilon_1 < T_{min3} - T_{max1}$, X_2 will be obtained and T_{max2} is its practical optimal cycle time. As $T_{max1} < T_{min3} < T_{max3} < T_{max2}$, the solution X_3 will be missed during the search process. In order to avoid missing possible local minima such as X_3, in the interval $\left[T_{k+1}, T_{max}^{X_{k+1}}\right]$, the gradient search approach, for example Frank-Wolfe method, may be employed. Details on the Frank-Wolfe method can be found in [5], [6] and [7].

Case 2. If T_{k+1} is infeasible, this means that we reach a sub-interval in the domain of T which contains no feasible solution (see for example Fig. 2.D). In this case, since the inventory holding cost is proportional to T in our particular problem. When $T_{k+1} = T_{max}^{X_k} + \epsilon_k$ is infeasible, a jump can be made to avoid the infeasible interval through the solution of the following cycle time relocating problem:

The Cycle Time Relocating Problem P2

Minimize

$$H(X,T) = \sum_{j \in S} \frac{1}{2} \eta_j q_j - \sum_{i \in S^+} \sum_{j \in S^+} \lambda_j x_{ij} \tag{13}$$

Subject to:

$$T_{max}^{X_k} + \epsilon_k \leq \sum_{i \in S^+} \sum_{j \in S^+} t_{ij} x_{ij} \leq T. \tag{14}$$

where $T_{max}^{X_k} + \epsilon_k$ is defined as above. Besides the modified constraints (14), other all constraints (4)-(5), (7)-(9) in $P1$ should also be taken into account. As a consequence, the solution X of $P2$ provides a feasible solution for $P1$. The obtained value of T is the minimal value for which a feasible solution X exists (*i.e.* $T = \sum_{i \in S^+} \sum_{j \in S^+} t_{ij} x_{ij}$). Once feasibility of T is reestablished again, the process continues as described in case 1.

To summarize, we outline the proposed solution strategy in the form of an algorithm as follows:

Let $[T_s, T_l]$ be the domain of T, XC be the set of corresponding feasible solutions X, and $\nabla f(X,T)$ be the set of the derivatives of objective function at X and T.

Algorithm 1. *(The main algorithm)*

Step 0. {*Initialization*}

Set $k = 0$ and choose a small $T_k \in [T_s, T_l]$ for which a feasible solution exists. Solve the problem P1, find a solution $X_k \in XC$.

Step 1. {*Checking the optimality*}

If $(T_k \geq T_l)$ *then stop.*

For the obtained solution X_k*, find the corresponding* $T_{max}^{X_k}$*. In* $\left[T_k, T_{max}^{X_k}\right]$*, call the Frank-Wolfe algorithm to find the optimal solution* $X_k^* \in XC$ *within this interval.*

Step 2. {*Updating T*}

Determine ϵ_k *and set* $T_{k+1} = T_{max}^{X_k} + \epsilon_k$ *and solve the problem P1. Set* $k = k+1$.

If (T_k *is infeasible*)

then (*solve the problem P2 and go to* Setp 1)

else (*go to* Setp 1)

The employed Frank-Wolfe algorithm is outline in the following algorithm:

Algorithm 2. *(The Frank-Wolfe algorithm)*

Step 0. {*Initialization*}

Set $n = 0$*, and choose* $(X_n, T_n) = (X_k, T_k)$.

Step 1. {*Generating the derivatives*}

Compute $\nabla f_n(X,T) = \left(\frac{\partial f}{\partial X}, \frac{\partial f}{\partial T}\right)|_{(X,T)_n}$.

Step 2. {*Solving a linear mixed integer minimal problem* $g(X,T)$}

Find an optimal solution X_{n+1}^{IP} *and* T_{n+1}^{IP} *for*

$$g(X,T) = Minimize\{\nabla f_n(X) \cdot X + \nabla f_n(T) \cdot T\}$$

.

Step 3. {*Generating* α}

Find $\alpha \in [0,1]$ *such that*

$$f\left(\alpha\left(X_{n+1}^{IP}, T_{n+1}^{IP}\right) + (1-\alpha)(X_n, T_n)\right) = Min\left\{f\left(\alpha\left(X_{n+1}^{IP}, T_{n+1}^{IP}\right) + (1-\alpha)(X_n, T_n)\right)\right\}$$

and set

$$(X_{n+1}, T_{n+1}) = \alpha\left(X_{n+1}^{IP}, T_{n+1}^{IP}\right) + (1-\alpha)(X_n, T_n)$$

.

Step 4. {*Updating*}

Choose (X_{n+1}, T_{n+1})*, set* $n = n+1$ *and go to* Setp 1.

The Frank-Wolfe algorithm stops when $\alpha = 1$.

The algorithm starts from the smallest value of T for which a feasible solution exits. It then updates T step by step avoiding the possibility of missing potential local minima. If an infeasible cycle time is found a cycle time relocating problem is solved to obtain a next new feasible cycle (a jump to the next smallest feasible cycle time). When the renewed T exceeds the feasible domain of the cycle time, the algorithm stops.

4 Numerical Example

A randomly generated simple problem is used to test the algorithm. The data are selected from the example in [2], which include the first 7 customers (*i.e.* namely $1, 2, ..., 7$, additional with the depot RW. See Fig.3.). The other all details are given in [2].

In this case, $T_s = 45.89h$, $T_l = 917.43h$. The starting cycle time is chosen to be $T = 50h$. AMPL CPLEX^R is used to solve the problem, the computational results are given below:

Table 1. Computational result

Parameters	Results
Optimal T	132.45h
Optimal X	(1, 3, 5), (6, 2), (4), (7)
Objective value (RC)	-303.39EUR/h
Cost rate	94.10EUR/h
CPU time	9640.62ms

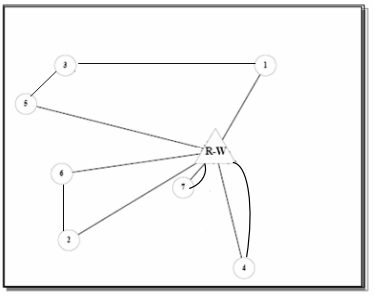


Fig. 3. The example figure for the solution

In Table 1, the cost rate is defined as:

$$\text{Cost rate} = \psi + \sum_{i \in S^+} \sum_{j \in S^+} \left((\delta \nu t_{ij} + \varphi_j) \frac{1}{T} + \frac{1}{2} \eta_j d_j T \right) x_{ij} \tag{15}$$

The procedure has actually found the optimal solution of problem. In this case, the out-loop has run 20 steps. In each in-loop, we needed only one step to determine the local minimum. However, we believe that for other more complicated cases, the in-loop will probably require more steps.

5 Conclusion

In this paper, the particular single-vehicle IRP is discussed. The analysis of problem revealed the difficulties that must be tackled if one whishes to solve

the problem to optimality. We proposed a steepest descend like hybrid solution strategy based on the gradient search to find the near-optimal solution. A numerical example is used to show the steps of the solution procedure. Some other cases of medium sizes were also solved with the procedure. In almost all cases the optimal solution is found. This approach, however, requires the solution of many MILP's. This results in a large computational time, which may be a serious drawback, especially for large-scale cases. The issue of locating those local minima quickly still needs to be addressed. Some extensions and other efficient search strategies are now under investigation.

References

1. Campbell, Melissa, A., Savelsbergh, M.W.P.: A Decomposition Approach for the Inventory-Routing Problem. Transportation Science 38(4), 488–502 (2004)
2. Aghezzaf, E.H., Raa, B., Van Landeghem, H.: Modeling Inventory Routing Problem in Supply Chain of High Consumption Products. European Journal of Operation Research 169, 1048–1063 (2006)
3. Aghezzaf, E.H.: Robust distribution planning for supplier-managed inventory agreements when demand rates and travel times are stationary. Journal of the Operational Research Society (2007)
4. Raa, B.: Models and Algorithms for the Cyclic Inventory Routing Problem. PhD thesis, Gent University, Belgium, pp. 16–48 (2006)
5. Frank, M., Wolfe, P.: An Algorithm for Quadratic Programming. Naval Research Logistics Quarterly 3, 95–110 (1956)
6. Chryssoverghi, I., Bacopoulos, A., Kokkinis, B., Coletsos, J.: Mixed Frank-Wolfe Penalty Method with Applications to Nonconvex Optimal Control Problems. Journal of Optimization Theory and Applications 94(2), 311–334 (1997)
7. ZhiQuan, L., SuZhong, Zh.: On Extensions of the Frank-Wolfe Theorems. Computational Optimization and Applications 13, 87–110 (1999)

A Methodology for the Automatic Regulation of Intersections in Real Time Using Soft-Computing Techniques

Eusebio Angulo, Francisco P. Romero, Ricardo García,
Jesús Serrano-Guerrero, and José A. Olivas

Escuela Superior de Informática,
Universidad de Castilla - La Mancha,
Paseo de la Universidad, 4, 13071, Ciudad Real, España
Eusebio.Angulo@alu.uclm.es, {FranciscoP.Romero,Ricardo.Garcia,
joseangel.olivas,Jesus.Serrano}@uclm.es

Abstract. This work presents an application of diverse soft-computing techniques to the resolution of semaphoric regulation problems. First, clustering techniques are used to discover the prototypes which characterize the mobility patterns at an intersection. A prediction model is then constructed on the basis of the prototypes found. Fuzzy logic techniques are used to formally represent the prototypes in this prediction model and these prototypes are parametrically defined through frameworks. The use of these techniques supposes a substancial contribution to the significance of the prediction model, making it robust in the face of anomalous mobility patterns, and efficient from the point of view of real-time computation.

Keywords: Regulating traffic lights, soft-computing, clustering, estimation models.

1 Introduction

The semaphoric regulation problem seeks to optimize i) the cycle lengths of a set of traffic-lights, ii) the percentage of time devoted to each of the phases in a cycle and iii) the transitions between consecutive sets of lights. This problem has been tackled in two temporal planning contexts. In the medium term, the stationary situation of the traffic is considered, and the objective is to obtain the semaphoric regulation of a set of intersections within the network. This problem has been formulated through a mathematical program with equilibrium constraints (MPEC). The results of these models are semaphoric regulations with fixed times for the cycles. The short term methods, which consider the dynamic aspect of the problem, have been fundamentally tackled through the application of optimization techniques to simulation models [1].

Various works using soft-computing techniques exist, and such works have fundamentally used Genetic Algorithms [2], whose objective has been the optimization of semaphoric transitions [3]. Numerous fuzzy logic approximations

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 379–388, 2008.

have also been carried out, particularly in the field of the fuzzy control of traffic-lights [4],[5], [1], [6] and [7]. Many of these developments have been carried out in an off-line context. The appearance of new traffic control technologies permits the real-time availability of precise data with regard to traffic conditions and makes the development of on-line methodologies possible.

Besides, Sanchez [8] presents architectures for traffic light optimization based on Genetic Algorithms with greater stability. It is designed and tested an evolutive architecture which optimizes the traffic light cycles in a flexible and adaptive way. These tests were of medium size and took place in a zone of Santa Cruz de Tenerife (Spain), thus improving the results of fixed cycle traffic lights.

In spite of these approximations, problems still remain which must be solved. One of these problems is that of tackling non-stationary mobility patterns, which is to say, the changing demands at various times of the day. This paper tackles this problem by proposing a methodology for the adaptive control of semaphoric intersections by using on-line traffic light counts.

The methodology here proposed is based on the extraction of mobility patterns on the basis of prototypes through the use of diverse soft-computing techniques which are implemented as an approach of the classic process of Knowledge Discovery on Databases (KDD) [9]. The use of diverse techniques, such as fuzzy logic and clustering, are incorporated in to this model and these techniques allow us to obtain more comprehensible and useful results for the prediction process.

The remainder of the work is organized as follows: Section 2 describes the different tasks that have been carried out to design the mobility patterns-based model. Section 3 explains the necessary stages to apply the above-designed model at a real intersection. To assess the methodology here proposed, an experiment has been developed in section 4. Finally, some conclusions and future works are pointed out.

2 Methodology

The objective of the methodology here proposed is that of adaptively regulating an intersection, as is shown in Figure 1. The intersection has sensors which measure all four lanes and permit the existence of entrance and exit traffic linkcounts in both directions at each of the time intervals considered. Moreover, that intersection has a semaphoric regulation.

To build the model with which to determine adaptive regulation, it is first necessary to extract the intersection´s mobility patterns. These patterns will be extracted from the vehicle flow observations obtained from the sensors.

The following stages are carried out to build the model:

1. Observations of the entrance/exit flows by use of the sensors.
2. Estimation model for *traffic dynamic O-D matrix*: This model permits the estimation of turns at the intersection. The O-D matrix is defined as being the matrix which contains, in the *i-j row*, the flow (number of people per time unit) which is incorporated into the intersection of lane i, and which

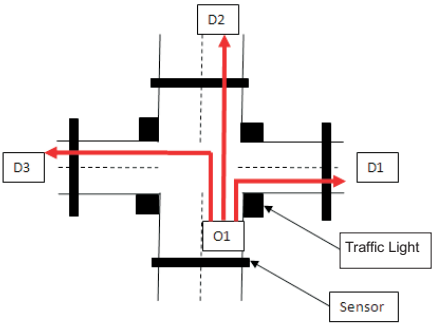


Fig. 1. Four lane intersection with no U-turns

leaves that lane via lane j. It is assumed that U-turn movements are not allowed, i.e., the main diagonal entries (i-i) of the O-D matrix are all zero.

3. Extraction of mobility Patterns: This stage models the mobility patterns using the O-D matrix and represents them by means of fuzzy deformable prototypes.
4. Traffic light regulation model: An expert can model the optimum behaviour of the semaphores following the above-mentioned prototypes.

In a concrete moment of the day, it can be seen different O-D matrices. The differences among themselves are random, so it can be considered that all matrices represent the same behaviour. So, such concept is here known as *mobility pattern* during a concrete time period and the exact representation of this pattern is each above-mentioned matrix.

2.1 Flow Observations

The entrance/exit sensors situated in each lane calculate the number of vehicles that are been driven in the instant t. We thus obtain the number of vehicles which pass each sensor, although their destinations are unknown owing to the turns that they may make.

Let an intersection be a tuple composed by m entrances and n exits and considering the time divided into N intervals ($t = 1, \ldots, N$), the inputs of the model would be:

- Entrance flows: $q_i(t)(i = 1, ..., m)$; $q(t) = [q_1(t), ..., q_m(t)]^T$;
- Exit flows: $y_j(t)(j = 1, ..., n)$; $y(t) = [y_1(t), ..., y_n(t)]^T$;

These data are the linkcounts in the intersection during the time period t. We thus obtain the number of vehicles which pass each sensor, but their destinations (turns) are unknown.

2.2 Estimation Model: Dynamic O-D Matrix

The estimation model shown in this sub-section is used to obtain the O-D matrix with complete predictions (including turns). This estimation can be carried out instantaneously by using the sensors information.

The model´s variables are the following:

- I_j : A set of values in which entrance i allows user to take exit j. I_j $(j = 1, \ldots, n)$.
- The probability that a vehicle enters via i and takes the exit j. $b_{ij}(i = 1, \ldots, m; j = 1, \ldots, n)$.
- The probability vector from each entrance i to the exit j. $b_j = [b_{ij}] \forall i \epsilon I_j$ y $Q_j = [q_i] \forall i \epsilon I_j$; $b = [b_1^T, ..., b_n^T]^T = [b^{(i)}]$

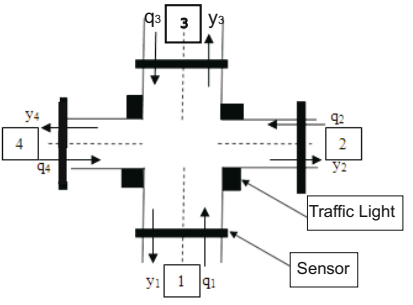


Fig. 2. q_i entrances and y_i exits in a four lane intersection with prohibited U-turns

For an intersection such as that shown in Figure 2, in which $n = m = 4$, the variables are as follows:
$b_1 = [b_{21}, b_{31}, b_{41}]^T$, $b_2 = [b_{12}, b_{32}, b_{42}]^T$, $b_3 = [b_{13}, b_{23}, b_{43}]^T$, $b_4 = [b_{14}, b_{24}, b_{34}]^T$
$Q_1 = [q_2, q_3, q_4]^T$, $Q_2 = [q_1, q_3, q_4]^T$, $Q_3 = [q_1, q_2, q_4]^T$, $Q_4 = [q_1, q_2, q_3]^T$

$b = [b_{21}, b_{31}, b_{41}, b_{12}, b_{32}, b_{42}, b_{13}, b_{23}, b_{43}, b_{14}, b_{24}, b_{34}]^T = [b^{(1)}, ..., b^{(12)}]^T$
Where b should fulfil:

$$b \geq 0, \sum_{j=1}^{n} b_{ij} = 1, b_{ii} = 0 \tag{1}$$

Therefore, the sum of probabilities from each entrance i to an exit j must be 1 and each probability must be greater than 0. For the observed linkcounts $y(t)$ and $q(t)$ in each time interval t, the estimation problem of b is resolved with:

$$J_i(b) = \sum_{s=1}^{t} \sum_{j=1}^{n} \{y_j(s) - Q_j(s) b_j\}^2, t = 1, ..., N \tag{2}$$

Where J_i is the set of values in which exit j permits users to enter entrance i, being $(i = 1, \ldots, m; j = 1, \ldots, n)$.

The estimation model creates an O-D matrix taking into account the turns. These vectors are the input of the phase called mobility patterns extraction.

2.3 Mobility Patterns Extraction

A clustering process is carried out to find relationships among the O-D matrices and after this process, the mobility patterns are detected. The goal of the

clustering process is to reduce the amount of data by categorizing or grouping similar data items together. Firstly the process must be build a similarity matrix based on the matrices returned by the estimation model, i.e., the inputs of the prototypes extraction process. The euclidean distance is the measure chosen to calculate the similarity among vectors.

Once the similarity matrix has been created, the two stages of the clustering process are carried out. Firstly the goal is finding groups of similar flows data detected in successive instants. This goal is reached following a graph-based clustering method [10]. In the second stage, to detect other similar groups that exist in non-successive instants, is carried out a hierarchical clustering algorithm based on fuzzy graph connectedness [11]. The nodes of the graph are the clusters of the first stage.

Every cluster represents a mobility pattern found at the intersection. Every pattern is described by a fuzzy deformable prototype that finally will be represented by a fuzzy numbers set. The fuzzy numbers set is modeled by a normalization and aggregation process using the O-D matrices of each cluster. This process permits to calculate the center and the length of the base of the fuzzy triangular numbers, the unique necessary data to represent each fuzzy number. In figure 3 are shown five prototypes that are the output of the clustering stage.

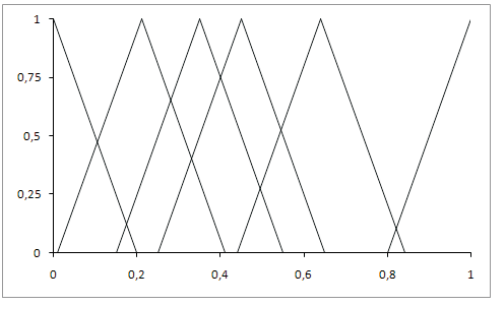


Fig. 3. Formal representation of the prototypes

Using a fuzzy numbers-based representation, it is easy to calculate the membership degree (in range [0-1]) between real situations and the prototypes detected.

2.4 Semaphoric Regulation

Once the mobility patterns have been detected and defined by means of fuzzy prototypes, the behaviour of each semaphore is analyzed by an expert depending on each pattern. If there are N prototypes then there are N optimum system responses and the value set of each response will be represented by a frame.

Fuzzy deformable prototypes and the parametric definition of the semaphoric regulation permits to design a flexible solution for the problem of traffic tie-ups. This idea could be especially important in critical moments such as great sport or cultural events, where the traffic can be a serious problem.

Table 1. Parametric Description of the prototypes

Prot	Congestion Level	Demand Direction
P1	Congested (High)	go
P2	Semicongested (Medium)	go
P3	Without congestion (Very Low)	go
P4	Semicongested (Medium)	return
P5	Without congestion (Very Low)	return

3 Model Performance

Once the model has been calculated, it can be applied to the daily management of the intersection. The regulation system´s entrance data will be the real-time flow observations, and the exits will express the type of regulation that must take place at each moment.

3.1 Real-Time Flow Observations

At an intersection, sensors located in every entrance/exit of the lanes catch information about the number of vehicles driving in every moment. These data feed the system to discover the optimum semaphoric regulation parameters.

3.2 Estimation Model

The estimation model permits us to obtain the complete O-D matrix (including turns) from the linkcount estimations. This is obtained in exactly the same manner as in the model construction phase (off-line). The elements and calculations specified in sub-section 2.2 will thus also be applicable in this step.

3.3 Inference in Prototypes

The mobility pattern is calculated using the values of the O-D matrix by means of an inference process based on the fuzzy deformable prototypes of the model. The algorithm is:

1. Normalization of the values of the entrance O-D vector.
2. Aggregate the normalized values (X value).
3. Calculate the membership degrees of each prototype represented by fuzzy numbers. To assess a concrete situation (Figura 4) is necessary relevant information. This relevant information is achieved by calculating an affinity degree with the prototypes(μ_i).

Once the membership degrees between the real environment and the prototypes of the model have been calculated, the definition of the prototype, that represents the optimum behaviour of the semaphores, must be returned.

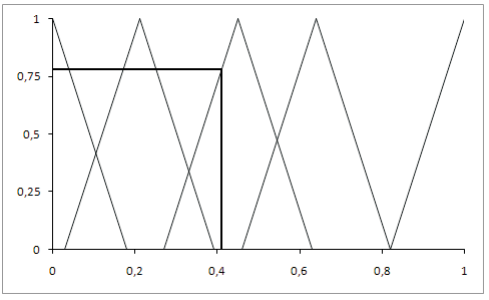


Fig. 4. Similitude of a vector to a prototype ($X = 0.41$, $\mu_3 = 0,78$)

3.4 Implementation of the Optimum Control

The most similar prototype among the above-calculated prototypes will be chosen as the most suitable one to simulate the semaphoric regulation in an exact moment.

The system´s exit thus contains all the parameters that define the traffic lights, behaviour whilst the detected mobility pattern remains. These values will be transmitted to the electronic component in charge of transmitting orders to each of the traffic lights in the intersection.

So, the output of the system is composed by all the parameters that are necessary to describe the behaviour of the semaphores while the mobility pattern is happening. The values of these parameters will be transmitted to the control process unit to manage the semaphores that are at the intersection.

4 Computational Experience

The data used in these numerical tests has been generated by simulation. The traffic density at each time interval is the same as that used in the demand which supports the urban railway network in Madrid. The graph in Figure 5 shows this hourly demand distribution.

Let q_i be the entrance traffic density in the approach i and e let y_i be the exiting traffic density in a determined time period. The estimation of the entrance flows q_i to the intersection is carried out by using the following expression:

$$q_i = D * p(t) * u(1 - \varepsilon, 1 + \varepsilon) \tag{3}$$

D: is the total entrance demand to the intersection, namely, the total number of vehicles passing through the intersection throughout the day and we consider 10000.
$p(t)$: is the proportion of turns dependant on the time instant. This parameter allows us to take into account the direction of the traffic flow in each instant.
$u(.)$: is the uniform random variable.
ε: takes the value of 0.15.

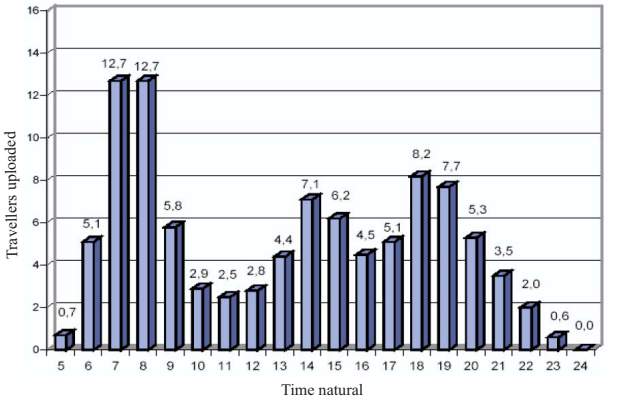


Fig. 5. Hourly demand distribution of the urban railway network, Madrid

The estimation of the exits y_i at the intersection is carried out by using the expression:

$$y_j = \sum_{j \neq i} (P_{ij}(t) * q_i) \tag{4}$$

$P_{ij}(t)$ is calculated by using the expresion:

$$P_{ij}(t) = \left(\frac{t-5}{24-5}\right) * P_1 + \left(1 - \frac{t-5}{24-5}\right) * P_2 \tag{5}$$

where P_1 and P_2 are:

$$P_1 = \begin{pmatrix} 0 & 0.2 & 0.6 & 0.2 \\ 0.1 & 0 & 0.5 & 0.4 \\ 0.2 & 0.4 & 0 & 0.4 \\ 0.25 & 0.25 & 0.5 & 0 \end{pmatrix} \quad P_2 = \begin{pmatrix} 0 & 0.4 & 0.2 & 0.4 \\ 0.5 & 0 & 0.25 & 0.25 \\ 0.6 & 0.2 & 0 & 0.2 \\ 0.5 & 0.4 & 0.1 & 0 \end{pmatrix}$$

Once the entrance and exit estimations for each lane have taken place for all the 5 minute time intervals between 05:00 and 24:00, the predicted origin-destination matrix is estimated by using the resolution of the proposed optimization model and by using GAMS software.

The estimation model allows us to obtain the complete O-D matrix (including turns) from the linkcount estimations and is calculated by using the elements and calculations specified in sub-section 3.2. Figure 6 shows the results of the linkcount estimation model, as opposed to those of the prediction model shown in sub-section 3.2, for the entrance turn in 1 and the exit in 2.

Figure 6 shows the results obtained. Note the high adjustment quality. This algorithm offers results which allow us to group the elements into 6 different mobility patterns. Figure 7 shows the different assignation of each element to the different groups obtained.

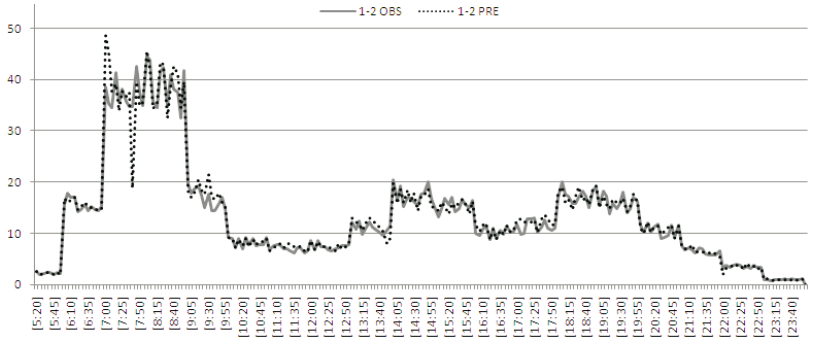


Fig. 6. Matrix $(1, 2)$ observed compared to the predicted matrix

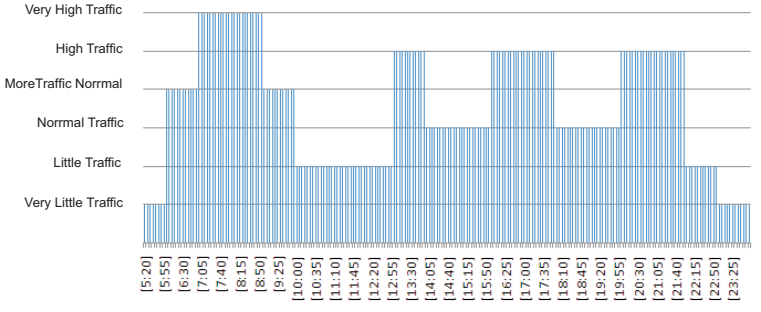


Fig. 7. Grouping distribution

5 Conclusions and Future Work

A new methodology has been presented to define and extract mobility patterns by means of optimization and fuzzy logic techniques. These techniques have been used to discover knowledge useful to design a formal, meaningful and useful model.

The methodology presents an automatic and adaptive control for intersections, achieving controlled outputs of the system and avoiding wrong responses. Besides, the requirements to develop a system based on these ideas are very simple due to the fact that the implementation of the system is really easy, the technology requirements are not expensive and the performance is very efficient.

To validate the proposed methodology, an experiment has been carried out simulating the behaviour of vehicles following known distributions. The performance of the experiment has been satisfactory.

In future works, the main goals are 1) testing a system developed following this methodology using real data and 2) refining the parameters used to described the mobility patterns.

Acknowledgments. This research has been partially supported by PAC06-0059 SCAIWEB project and PCC08-0081-4388-2, JCCM, Spain, TIN2007-67494 F-META project, MEC-FEDER, Spain, and FIT-340001-2007-4 BUDI project, MICYT, Spain.

References

1. Wiering, M., Vreeken, J., Van Veenen, J., Koopman, A.: Simulation and optimization of traffic in a city. In: IEEE Intelligent Vehicles Symposium, Proceedings, pp. 453–458 (2004)
2. Rouphail, N., Park, B., Sacks, J.: Direct signal timing optimization: Strategy development and results. In: XIth Pan American Conference on Traffic and Transportation Engineering (2000)
3. Lim, G.Y., Kang, J.J., Hong, Y.S.: The optimization of traffic signal light using artificial intelligence. In: Proc. 10th IEEE Int. Conf. Fuzzy Syst., December 2-5, 2001, vol. 3, pp. 1279–1282 (2001)
4. Lei, C., Guojiang, S., Wei, Y.: The traffic flow model for single intersection and its traffic light intelligent control strategy. In: Proceedings of the World Congress on Intelligent Control and Automation (WCICA) 2, art. No. 1713650, pp. 8558–8562 (2006)
5. Van Leeuwaarden, J.S.H.: Delay analysis for the fixed-cycle traffic-light queue. Transportation Science 40(2), 189–199 (2006)
6. Lim, G.Y., Kang, J.J., Hong, Y.S.: The optimization of traffic signal light using artificial intelligence. In: IEEE International Conference on Fuzzy Systems, vol. 3, pp. 1279–1282 (2002)
7. Hoyer, R., Jumar, U.: Fuzzy control of traffic lights. In: IEEE International Conference on Fuzzy Systems, vol. 3, pp. 1526–1531 (1994)
8. Sánchez, J., Galán, M., Rubio, E.: Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: "Las Ramblas" Area in Santa Cruz de Tenerife. IEEE Transactions on evolutionary computation 12(1), 25–40 (2008)
9. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The KDD Process for Extracting Useful Knowledge from Volumes of Data. Communications of the ACM 39(11), 27–34 (1996)
10. Kawaji, H., Yamaguchi, Y., Matsuda, H., Hashimoto, A.: A Graph-Based Clustering Method for a Large Set of Sequences Using a Graph Partitioning Algorithm. Genome Informatics 12, 93–102 (2001)
11. Dong, Y., Zhuang, Y., Chen, K., Taib, X.: A hierarchical clustering algorithm based on fuzzy graph connectedness. Fuzzy Sets and Systems 157, 1760–1774 (2006)

Composite Dispatching Rule Generation through Data Mining in a Simulated Job Shop

Adil Baykasoğlu[1], Mustafa Göçken[1], Lale Özbakır[2], and Sinem Kulluk[2]

[1] Gaziantep University, Department of Industrial Engineering, Gaziantep, Turkey
[2] Erciyes University, Department of Industrial Engineering, Kayseri, Turkey
{baykasoglu,mgocken}@gantep.edu.tr, {lozbakir,skulluk}@erciyes.edu.tr

Abstract. In this paper, a new data mining tool which is called TACO-miner is used to determine composite Dispatching Rules (DR) under a given set of shop parameters (i.e., interarrival times, pre-shop pool length). The main purpose is to determine a set of composite DRs which are a combination of conventional DRs (i.e., FIFO, SPT). In or-der to achieve this, full factorial experiments are carried out to determine the effect of input parameters on predetermined performance measures. Afterwards, the data set which is obtained from the full factorial simulation analyses is feed into the TACO-miner in order to determine composite DRs. The preliminary verification study has shown that composite DRs have an acceptable performance.

Keywords: Dispatching rules, data mining, simulation, ant colony optimization, neural networks.

1 Introduction

The computational complexity of job shop scheduling has stimulated interest in heuristics, meta-heuristics and other algorithms for solving large job shop problems in a reasonable computational time. One class of heuristics includes DR. DR are often favored because of their simplicity, ease of application and the fact that they are an on-line scheduling method that can be used in real-time scheduling. This makes them dynamic in the sense that they can process new job arrivals and react to other disruptions without need to re-schedule [1]. In this sense, DRs work well for dynamic scheduling problems. However, in the context of conventional job shops, the relative performance of these rules has been found to depend upon the system attributes, and no single rule is dominant across all possible scenarios. This indicates the need for developing a scheduling approach which adopts a state-dependent DR selection policy [2]. By this way, still the dispatching heuristic does not adapt itself to the changing system attributes but adoption of the most appropriate dispatching heuristic by switching between a set of pre-determined DRs according to the current state of the shop is provided. Thus, the deterioration of the selected performance measure can be prevented. In fact, this problem continues to be a very active area of research [3].

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 389–398, 2008.

Researches into DRs appear to take two main directions. The first uses simulation to evaluate new and existing DRs under different shop conditions and performance objectives [4]. The second direction aims to improve the performance of existing rules by using new tools or strategies to support or hybridize existing rules [1]. Approaches related to second category attracts many researchers and successful studies exist in the literature. For example, Pierreval and Mebarki [6] proposed a scheduling strategy which is based on a dynamic selection of certain pre-determined DRs. El-Bouri and Shah [7] proposed an intelligent system that selects DRs to apply locally for each machine in a job shop. Holthaus and Rajendran [8] developed new scheduling rules based upon the combination of well-known rules. Li and Olafsson [9] presented that by using decision tree models to learn from properly prepared data set, not only a predictive model that can be used as a DR can be obtained, but they also showed that previously unknown structural knowledge can be obtained that provides new insights and may be used to improve scheduling performance. Wang et al. [10] developed a hybrid knowledge discovery model, using a combination of a decision tree and a back-propagation neural network, to determine an appropriate DR for use in the semiconductor final testing industry in order to achieve high manufacturing performance. More recently, Baykasoğlu et al. [11] proposed a new data mining approach that is known as MEPAR-miner (Multi-expression programming for classification rule mining) is used to extract knowledge on the selection of best possible DRs among certain pre-determined DRs. This study applies a new data mining approach that is known as TACO-miner for deriving composite DRs.

2 Problem Definition

This study aims at developing a composite DR which optimizes a specific performance measure under changing input parameter levels (i.e., arrival rate, buffer size etc.). By this aim, first, factors (input parameters) determined in order to identify the behavior of the system under different levels of input parameters. Mean absolute percentage error (MAPE) is selected as the system's response variable. Since sophisticated rules hardly provide significant improvement over most less sophisticated ones, simple and easy to implement DRs was selected [12]. The selected DRs were first in first out (FIFO), shortest processing times (SPT), and earliest due date (EDD). The present problem is to find classification rules which can classify different factor levels according to selected DRs by using the TACO-miner algorithm.

3 A Brief Overview of the TACO-Miner Algorithm

Many approaches have been proposed in the literature so far in order to develop effective algorithms for classification rule extraction. In the past, artificial neural networks (ANN) was used commonly and found to be one of the most efficient

tools for classification and prediction purposes. However, they have the well-known disadvantage of having black-box nature and not discovering any high level rule that can be used as a support for human understanding. Because of that, many researchers tend to develop new algorithms for rule extraction from ANNs. The knowledge acquired by an ANN is codified on its connection weights, which in turn are associated to both its architecture and activation functions [14]. In this context, the process of knowledge acquisition from ANNs usually implies the use of algorithms based on the values of either connection weights or hidden unit activations. The algorithms designed to perform such task are generally called algorithms for rule extraction from ANNs [13]. Recently Özbakir et al. [14] proposed such a new rule extraction algorithm from ANNs. The proposed algorithm is mainly based on a meta-heuristic which is known as Touring Ant Colony Optimization (TACO) and has a two-step hierarchical structure. In the first step a multilayer perceptron type ANNs is trained and its weights are extracted. After obtaining the weights, in the second step TACO is applied to extract classification rules.

In this study, multi-layer perceptron (MLP) which is one of the most widely used ANN is considered. For extracting classification rules from trained MLP via TACO algorithm, elements of the dataset must be decoded in binary form. For the data sets which contain continuous attributes, discretization must be carried out before transforming the attributes into binary form. MLP is trained on the encoded vectors of the input attributes and the corresponding vectors of the output classes until the convergence rate between actual and the desired output is achieved. The general methodology of rule extraction from ANNs by TACO is shown in Figure 1 taken from Özbakır et al. [13]. Due to space limitation the details of TACo algorithm is not given here. However, for details of the algorithm the reader is referred to Özbakır et al. [13].

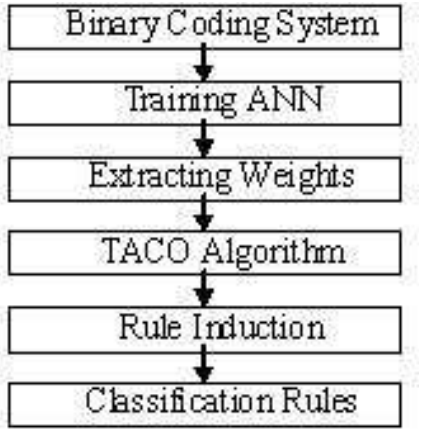


Fig. 1. General flowchart for the proposed methodology

4 Simulation Study

Simulation experiments are conducted on a hypothetical multi-stage dynamic job shop to collect data for different levels of input parameters and response variable. The job shop model is the same as in Baykasoğlu et al. [11]. For convenience we define the model as follows:

4.1 Job Shop Model

The simulation model represents a multi-stage job shop containing 24 workstations. There is only one machine in each workstation. The interarrival times are assumed to be exponentially distributed. The system observed for three different levels of the interarrival time attribute (see table 1). The total number of operations of a job varied uniformly from 1 to 10 and assigned at the arrival time of the job by generating a random number between 1 and 10. Also, routes of the jobs are determined according to the uniformly determined operation sequence For example, if total number of operations of an arriving job assigned to be 3, a random number between 1 and 24 is generated and assigned as its first operation no and a second one generated and assigned as its second operation no, and the third one generated and assigned as its third operation no. Each machine can process only predetermined operations and also one operation at a time (see Table 2). A given machine could appear more than once in a job's routing even for consecutive operations. The processing times of each job are assumed to be uniformly distributed and given in Table 2. Transportation times are assumed to be negligible. The simulation model is run for 18000 finished jobs and after preliminary runs 8000 time unit is determined to be warm-up period to avoid incorporation of transient behavior of the system. The experiments are performed using 5 replications of each selected DR and proposed composite DR, thus minimizing variability in the results. Common random numbers are used to provide the same experimental condition across the runs for each factor combination.

All arriving jobs are waited in an entrance pool according to the FIFO DR. In the case of exceeding the capacity of entrance pool, arriving jobs are rejected (see Table 1 for entrance pool's capacity levels). Due dates of jobs are assigned by using conventional total work content (TWK) due date assignment rule. Three levels of due date tightness, tight, medium, and loose due dates are determined, by letting due date tightness factor differentiate between 30, 50, and 75 (see Table 1), respectively. Before releasing a job to the shop floor, the entrance pool is searched. The search is made according to the 'search dept of queue' (SDEPQ) attribute (see Table 1 for levels of SDEPQ attribute). For example, if the level of SDEPQ is determined to be 25, only first 25 jobs from entrance pool will be checked whether there exist any mature jobs. In the case of being mature they released to the shop floor otherwise they kept waiting in the entrance pool. For being a mature job, the number of waiting jobs in each machine's queue placed in the job's route must be lower than 'buffer size' attribute's value (BUFSIZ) (see table 1 for levels of BUFSIZ attribute). If a job is found to be not mature it is not released to the shop floor and SDEPQ attribute's value

Table 1. Factors and their levels

Factors	Levels		
Interarrival Times (INTARR) (unit time)	1.5	3	4.5
Entrance Pool Length (EPL) (jobs)	1000	3000	5000
Search Depth of Queue (SDEPQ) (jobs)	25	50	75
Buffer Size (BUFSIZ) (jobs)	50	75	100
Due Date Tightness (DDTIGHT) (constant)	30	50	75

Table 2. Processing times of operations according to stations

Operation No	*Station No*	*Process Time*	*Op. No*	*St. No*	*Pr. Time*	*Op. No*	*St. No*	*Pr. Time*
1	9	U(15,20)	9	7	U(10,15)	17	13	U(10,20)
2	10	U(12,18)	10	3	U(15,25)	18	14	U(10,15)
3	20	U(12,18)	11	11	U(15,20)	19	15	U(10,15)
4	21	U(12,18)	12	2	U(10,15)	20	18	U(12,18)
5	17	U(15,20)	13	4	U(12,20)	21	22	U(10,15)
6	16	U(10,15)	14	5	U(15,20)	22	23	U(15,20)
7	1	U(15,20)	15	6	U(12,18)	23	24	U(10,15)
8	8	U(12,18)	16	12	U(15,20)	24	19	U(15,20)

is increased by 1. If, SDEPQ value is lower than its predetermined level (i.e., 25), checking for subsequent jobs is performed. Simulation model is run for all input parameter combinations and data related to MAPE performance measure is gathered. Consequently, the test and training data are ready for rule extraction and verification of the extracted rules.

4.2 Simulation Experiments

Experiments are performed according to the full factorial design in order to detect the effects of input parameters on the performance criterion accurately. Factors of experimental design and their levels are shown in Table 1. Since there are 5 factors, 3 levels for each, 243 (35) experiments are performed. These experiments are repeated for FIFO, SPT and EDD DRs in order to determine the selected performance measure's values as the response variable's value desired to be minimized. Parameter settings and predictive accuracies of TACO-miner algorithm and parameter settings of ANN and on the data set are summarized in Table 3, Table 4, and Table 5, respectively.

5 The Methodology and Results

After generating data for different levels of input parameters and response variable, these are used as input to multi-layer perceptron (MLP) ANN for training purpose. MLP is trained on the encoded vectors of the input attributes and

Table 3. Parameter setting of TACO-miner algorithm

No.of Ants(M)	*No.of Iterations*(T)	*Frequency Factor*(f)	*Evaporation Parameter*(ρ)	*ConstantQ*
100	1000	2	0.8	5

Table 4. Predictive accuracies of TACO-miner on the data set

Performance metric	Min	Average	Max	Standard Deviation
Testing Accuracy (%)	95.06	96.9	98.77	1.674
Extracted number of rules	8	10.5	13	1.149

the corresponding vectors of the output classes until the convergence rate between actual and the desired output is achieved. After training the MLP, weights from input layer to hidden layer and weights from hidden layer to output layer are extracted by using sigmoid activation function. Then, these weights are fed into TACO-miner algorithm for rule induction process. In this stage, TACO-miner uses weights to extract rules belonging to certain classes. Applying this methodology, a composite DR for MAPE performance measurement was extracted. Here, the aim is to develop a system state-independent composite DR which is impossible for conventional DR in a dynamic job shop environment. This means, developed composite DR always provides acceptable MAPE performance for all levels of system attributes. Developed composite DR is given in table 6.

Composite DR is tested with 10 randomly selected input parameter combinations under the same experimental conditions. For each selected input factor combination. Results of the composite and conventional DRs with respect to MAPE performance measurement for each factor combinations are summarized in table 7.

Simulation experiments revealed that according to the MAPE performance, composite DR performed very well for all possible input parameter combination. Note that, for only 3 system combinations (3, 4, and 9) composite DR's performance is worse than the conventional DRs, but is acceptable. In figure 1, performances of DRs shown comparatively for each input parameter combination. From figure 1, it is apparently seen that under all possible input parameter combination composite DR can be used confidently for achieving acceptable MAPE performance.

Table 5. Parameter setting of ANN

Data Set	*#of Hidden Layer*	*Proc. Elementsin HiddenLayer*1	*Proc. Elementsin HiddenLayer*2	*Transfer Function*	*Max Epoch*	*Learning Rule*
Job Shop	2	12	5	Sigmoidaxon	20000	*Conjugate gradient*

Table 6. Developed composite DR by TACO-miner

$Rule1 : IF, (INTARR = 45)AND((SDEPQ = 50)OR(SDEPQ = 75))AND$ $(BUFSIZ = 75)AND((DDTIGHT = 50)OR(DDTIGHT = 75)), FIFO :$
$Rule2 : IF, (INTARR = 15)AND(PRL = 5000)AND(DDTIGHT = 30), SPT :$
$Rule3 : IF, (INTARR = 15)AND((PRL = 3000)OR(PRL = 5000))AND$ $((BUFSIZ = 75)OR(BUFSIZ = 100))AND(DDTIGHT = 30), SPT :$
$Ruel4 : IF, (INTARR = 15)AND((PRL = 3000)OR(PRL = 5000))AND$ $((SDEPQ = 25)OR(SDEPQ = 75))AND(DDTIGHT = 30), SPT :$
$Rule5 : IF, (INTARR = 3), EDD :$
$Rule6 : IF, ((SDEPQ = 25)OR(SDEPQ = 75))AND((BUFSIZ = 50)OR$ $(BUFSIZ = 100))AND((DDTIGHT = 50ORDDTIGHT = 75)), EDD :$
$Rule7 : IF, ((INTARR = 15)OR(INTARR = 3))AND((DDTIGHT = 50)OR$ $(DDTIGHT = 75)), EDD :$
$Rule8 : IF, ((INTARR = 15)OR(INTARR = 3))AND((PRL = 1000)OR$ $(PRL = 3000))AND((SDEPQ = 50)OR(SDEPQ = 75)), EDD :$
$Rule9 : IF, (INTARR = 45)AND((BUFSIZ = 50)OR(BUFSIZ = 75))AND$ $(DDTIGHT = 75), FIFO :$
$Rule10 : IF, ((BUFSIZ = 50)OR(BUFSIZ = 100))AND((DDTIGHT = 50)OR$ $(DDTIGHT = 75)), EDD :$
$Rule11 : IF, (DDTIGHT = 50), EDD :$
$Rule12 : IF, (INTARR = 15)AND((SDEPQ = 25)OR(SDEPQ = 50))AND$ $((BUFSIZ = 75)OR(BUFSIZ = 100))AND(DDTIGHT = 30), SPT :$
$Rule13 : IF, (INTARR = 45)AND((BUFSIZ = 50)OR(BUFSIZ = 100))AND$ $(DDTIGHT = 30), FIFO :$
$Rule14 : IF, (INTARR = 45)AND((SDEPQ = 25)OR(SDEPQ = 50))AND$ $(DDTIGHT = 75), FIFO :$
$Rule15 : IF, (INTARR = 45)AND(SDEPQ = 25)AND((DDTIGHT = 30)OR$ $(DDTIGHT = 75)), FIFO :$
$Rule16 : IF, (INTARR = 15)AND((PRL = 3000)OR(PRL = 5000))AND$ $(SDEPQ = 50)AND = 30)$ $((BUFSIZ = 75)OR(BUFSIZ = 100))AND((DDTIGHTOR(DDTIGHT = 75)), SPT :$
$Rule17 : IF, ((SDEPQ = 25)OR(SDEPQ = 75))AND((BUFSIZ = 50)OR$ $(BUFSIZ = 100)), EDD :$
$Rule18 : IF, ((SDEPQ = 25)OR(SDEPQ = 75))AND((BUFSIZ = 75)OR$ $(BUFSIZ = 100)), EDD :$
$Rule19 : ELSE, EDD;$

Table 7. Results for MAPE (%)

InputParameter Combinations	IntArr	Prl	SDEPQ	BufSiz	DDTight	FIFO	EDD	SPT	*Composite Rule*
1	1.5	1000	25	50	30	0.11	0.10*	0.11	0.10*
2	1.5	1000	50	75	50	0.10	0.08*	0.12	0.08*
3	1.5	1000	75	100	30	0.15	0.14*	0.15	0.1
4	1.5	3000	50	100	30	0.28	0.28	0.26*	0.28
5	1.5	5000	25	50	75	0.25	0.24	0.25	0.23*
6	3	3000	25	100	30	0.14*	0.14*	0.17	0.14*
7	4.5	1000	25	50	30	0.05*	0.06	0.06	0.05*
8	4.5	1000	50	100	75	0.17	0.17	0.17	0.16*
9	4.5	3000	50	50	75	0.16*	0.17	0.17	0.17
10	4.5	5000	50	100	30	0.06*	0.06*	0.06*	0.06*

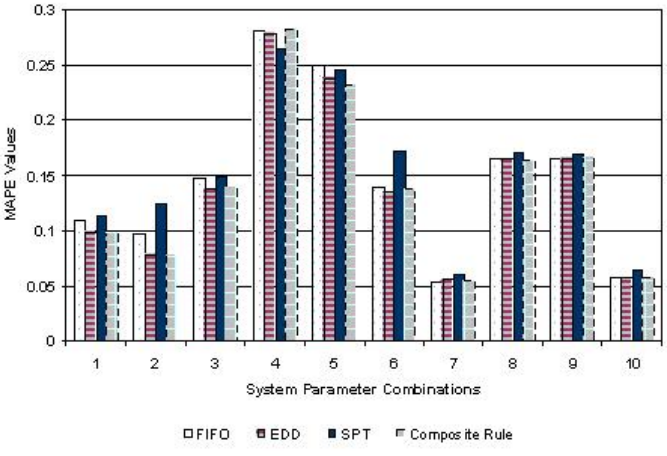


Fig. 2. Comparative performances of DRs

6 Conclusions

Conventional DRs' performances are significantly affected by the state of the system which changes frequently in job shop environments. In such environments, selecting the most appropriate conventional DR in order to achieve the best performance is a challenging task. This is mainly because they are problem dependent. Thus, most of the time they can not adapt themselves to changing levels of input parameters which results in poor performance. One way of improving the performance of dispatching process is to use simple and easy to implement conventional DRs interchangeably according to the certain system state.

In literature, this problem tackled by using ANN. In most of the studies it is pointed out that, ANN's choice of DR resulted in better satisfaction of

the performance criteria than its counterparts. In fact, they achieved this by their high classification accuracy. But the main problem with them was their having a black-box nature which causes discovery of low-level rule that can not be used as a support for human understanding. The recent trend is to acquire comprehensible knowledge from trained ANNs. In this manner, high-level rules which can be used as a support for human understanding is extracted.

In this study, accurate and comprehensible composite DR in a dynamic job shop environment is developed. Before, developing such a composite DR, full factorial experiments are carried out to determine the effect of input parameters on predetermined performance measures for each selected conventional DR. The effects of input parameters are determined by training MLP ANN. In fact, the weights of MLP represent the effect of input parameters. Then, this knowledge is utilized by TACO-miner. By using this knowledge, TACO-miner extracted not only accurate but also comprehensible composite DR which is valid for all possible input parameter combinations. Developed composite DR is high-level which means it can be used to support for human understanding. In other words, by interpreting this composite DR, previously unknown structural knowledge can be obtained that provides new insights and may be used to improve scheduling performance. Simulation results revealed that the composite DR performs comparatively better than the conventional DRs in terms of MAPE performance measure. This is mainly due to the capability of composite DR in selecting the best possible DR according to the current system state.

Acknowledgments

Prof. Dr. Adil Baykasoğlu is grateful to Turkish Academy of Sciences (TÜBA) for supporting his scientific studies.

References

1. Al-Turki, U., Andijani, A., Arifulsalam, S.: A new dispatching rule for the stochastic single-machine scheduling problem. Simulation-Transactions of the Society for Modeling and Simulation International 80(3), 165–170 (2004)
2. Shaw, M.J., Park, S., Raman, N.: Intelligent scheduling with machine learning capabilities: The in-duction of scheduling knowledge. IIE Transactions 24(2), 156–168 (1992)
3. Jones, A., Rabelo, L.C.: Survey of job shop scheduling techniques, National Institute of Stan-dards and Technology Manufacturing Engineering Laboratory: Publications, (accessed September 2007), http://ws680.nist.gov/mel/div820/pubstracking/search.asp
4. Kiran, A.S., Smith, M.L.: Simulation studies in job shop scheduling-II: performance of priority rules. Computers & Industrial Engineering 8(2), 95–105 (1984)
5. Pierreval, H., Mebarki, N.: Dynamic selection of dispatching rules for manufacturing system sched-uling. International Journal of Production Research 35(6), 1575–1591 (1997)

6. El-Bouri, A., Shah, P.: A neural network for dispatching rule selection in a job shop. International Journal of Advanced Manufacturing Technology 31(3-4), 342–349 (2006)
7. Holthaus, O., Rajendran, C.: New dispatching rules for scheduling in a job shop–An experimen-tal study. International Journal of Advanced Manufacturing Technology 13(2), 148–153 (1997)
8. Li, X., Olafsson, S.: Discovering dispatching rules using data mining. Journal of Scheduling 8(6), 515–527 (2005)
9. Wang, K.J., Chen, J.C., Lin, Y.S.: A hybrid knowledge discovery model using decision tree and neural network for selecting dispatching rules of a semiconductor final testing factory. Production Planning & Control 16(7), 665–680 (2005)
10. Baykasoğlu, A., Göçken, M., Özbakır, L.: A Data Mining Approach to Dispatching Rule Selection in a Simulated Job Shop. In: Geiger, M.J., Habenicht, W. (eds.) Proceedings of EU/ME 2007 Meta-heuristics in the Service Industry, 8th Workshop of the EURO Working Group EU/ME, the European Chapter on Metaheuristics, Stuttgart, Germany, October 4-5, 2007, pp. 65–71 (2007)
11. Anderson, E.J., Nyirenda, J.C.: Two new rules to minimize tardiness in a job shop. International Journal of Production Research 28(12), 2277–2292 (1990)
12. Özbakir, L., Baykasoğlu, A., Kulluk, S.: Rule Extraction from Neural Networks via Ant Col-ony Algorithm for Data Mining Applications. In: Learning and Intelligent OptimizatioN -LION 2007 2nd International Conference, Trento, Italy, December 8-12 2007. LNCS. Springer, Heidelberg (in press, 2007)
13. Andrews, R., Diederich, J., Tickle, A.B.: A Survey, Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks. Knowledge Based Sys. 8(6), 373–389 (1995)
14. Hruschka, E.R., Ebecken, N.F.F.: Extracting Rules from Multilayer Perceptrons in Classi-fication Problems: A Clustering-based Approach. Neurocomputing 70, 384–397 (2006)

Co-author Network Analysis in DBLP: Classifying Personal Names

Maria Biryukov

University of Luxembourg,
6, rue Richard Coudenhove-Kalergi,
1359 Luxembourg, Luxembourg
maria.biryukov@uni.lu

Abstract. In this paper we describe how the co-author network, which is built from the bibliographic records, can be incorporated into the process of personal name language classification. The model is tested on the DBLP data set. The results show that the extension of the language classification process with the co-author network may help to refine the name language classification obtained from the author names considered independently. It may also lead to the discovery of dependencies between the elements of the co-author network, or participation of authors in scientific communities.

Keywords: language classification, bibliographic databases, co-author networks.

1 Introduction

With the constant growth of scientific publications, bibliographic databases and digital libraries become widespread. Services such as *CiteSeer* [8], *Google Scholar* [2], or *DBLP bibliography* [10], are often consulted to find publications in a given domain or identify people working in the area of interest. At the same time they are an object of research in their own right. The information contained in the bibliographical databases and digital libraries is being explored to gain insight into various aspects of scientific world. Take for example co-author networks which are built from the bibliographic records. Their analysis (sometimes in combination with more factors, such as time of publication, keywords in publication and/or venue titles) starts from calculating the "central" author for a certain venue [17] and extends to community discovery [21,20], understanding of scientific collaboration and individual researcher profiling [12,5,1], and topic modeling [18,11]. They are employed in data visualization tasks [14,6,7] and for the purpose of name spelling correction [13,3,9]. Although the examples above suggest a wide scope of directions in bibliographic data analysis, the question of *"where the authors come from"?* does not seem to attract much attention so far. In [4], an attempt to capture the geographical background of the papers published in SIGIR[1] conferences is reported. This analysis is

[1] SIGIR – Special Interest Group on Information Retrieval [16].

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 399–408, 2008.

based on the bibliographical data contained in the papers. However the personal name itself may shed light on the author's origin. In [22], we proposed a tool for language detection of personal names, and applied it to the set of more than 600,000 names recorded by DBLP. Each name was considered on an individual basis. We showed how such tool could be used in the process of data cleaning, namely in selection of the correct name spelling when multiple variations of the same name existed. In another experiment, the system was employed to discover how the share of participation of different cultures in scientific publications was evolving within the last 20 years. While these experiments proved the usability of the name language detection system for the bibliographic databases and digital libraries, they revealed a high number of names which could not unambiguously be attributed to one language and thus affected the success rate of the tool. Consider for example the name "John Li": the first component suggests English, while the second one points to Chinese. In order for such names (*mixed names* thereafter) to be classified correctly additional knowledge is required. It could eventually be obtained from the external sources, for instance personal homepages or institute affiliations. Alternatively we turn to the examination of co-author networks to solve the problem of language assignement.

This paper is organized as follows: in Section 2 we present the tool for the name language detection and introduce its application to the DBLP. Section 3 describes the name language classification approach enhanced with the co-author network analysis. Evaluation of the results is presented in Section 4. Finally we conclude the paper by a short summary of the results in Section 5.

2 Detecting the Language of a Personal Name in DBLP

2.1 System Overview

The language detection system we have built, consists of a set of corpora and a set of metrics for the estimation of the probability that a character string A belongs to a language L. While the tool is applied to the personal name language detection, the string A is not limited to represent a name. Rather it can be any valid string in some language L. The overlapping n-gram[2] model is chosen to represent both – the corpora and the names to be labeled. It is based on the assumption that the n-grams and their frequencies are specific to any given language and thus may serve as discriminating feature. The n-gram model of the language can be traced back to Shannon [15].

The system is trained to identify 14 different languages: Chinese, Danish, Dutch, English, Finnish, French, German, Italian, Japanese, Norwegian, Portuguese, Spanish, Swedish, and Turkish. For checking whether a string of characters $A = [a_0, a_1, \ldots, a_{l-1}]$ belongs to the language L we use the following formula:

[2] The term n-gram refers to the sequence of n characters, where $n \geq 1$. The word *overlapping* indicates that the last and the first characters of the $k_{th} - 1, k_{th}$ n-grams are the same.

$$P(A \in L) = p_L(a_0, a_1, a_2, a_3) \cdot \prod_{i=1}^{l-4} p_L(a_{i+3}|a_i, a_{i+1}, a_{i+2}).$$

Here, the probability p_L that the tetragram a_0, a_1, a_2, a_3 belongs to the language L is approximated by its frequency in the corpus of the language L, divided by the size M of the corpus:

$$p_L(a_0, a_1, a_2, a_3) \approx fr_L(a_0, a_1, a_2, a_3)/M$$

and conditional tetragram probability is approximated as follows:

$$p_L(a_{i+3}|a_i, a_{i+1}, a_{i+2}) \approx \frac{fr(a_i, a_{i+1}, a_{i+2}, a_{i+3})}{fr(a_i, a_{i+1}, a_{i+2})}.$$

If we denote by $logFr$ the logarithms of the frequencies and normalize the result by the length of the string l, we get:

$$\log P(A \in L) = logFr(a_0, a_1, a_2, a_3) - \log M +$$

$$+ \sum_{i=1}^{l-4} logFr(a_i, a_{i+1}, a_{i+2}, a_{i+3}) - logFr(a_i, a_{i+1}, a_{i+2}).$$

$$CondTetrScore(A) = \frac{logP(A \in L)}{l}.$$

This metric estimates the probability that the string A belongs to the language L using the conditional tetragram approximation of the language.[3]

Tetragrams which occurred < 3 times in the corpus are not considered, since their frequency may not be a good approximation of the real tetragram probability. For the n-grams that cannot be found or are infrequent in the language the default solution is to evaluate their weight to -1000 ("penalty"). It might be the case though that the corpus for that language is not sufficiently large to include all the possible n-grams that may occur in the names. Instead of assigning such n-grams the penalty weight immediately, we check whether the n-gram exists with a certain minimal frequency in the other languages. (We checked the range 100 – 10000 and stopped at the choice of the threshold frequency 100 as the one performing the best.) If the n-gram is sufficiently frequent in at least one of the other languages, we give the penalty weight in the language which is currently being checked. This way we increase the discriminating power of the computational model. Alternatively, the n-gram is approximated by an $(n-1)$-gram (for example for the tetragrams):

$$\log P(a_{i+3}|a_i, a_{i+1}, a_{i+2}) \approx$$

$$\approx logFr(a_{i+1}, a_{i+2}, a_{i+3}) - logFr(a_{i+1}, a_{i+2}).$$

[3] The conditional trigram approximation as well as unconditional models have also been tried. They are discussed in detail in [22].

At this stage however if the $(n-1)$-gram is not found in the language the conditional tetragram is penalized.

Our system is built in a way which allows an easy addition of new languages and new string evaluation metrics. The program takes as input a list of personal names for which their language origin has to be identified, and the parameter, which indicates the choice of the string evaluation metric [4]. The system outputs separate files with the names attributed to each language, ranked by the metric of choice. For each name the second best choice is given as well as the values of the metric across all the languages.

The tool has been tested on 100 names for each of the 14 languages as well as on the joint list of 1400 names, all collected from the Wikipedia people lists [19]. The first setting allows us to accurately assess the recall and precision achieved by the system when given a monolingual set of names. The second setting approximates the "real life" conditions of a database with a multilingual set of names. The overall performance has shown recall above 80% and precision in the range of 80 – 100% for most of the languages. These figures have motivated us to apply it to the DBLP.

2.2 Application to DBLP

DBLP is a publicly available database which provides bibliographic information on major computer science journals and proceedings. The records typically list the (co-)author name(s), publication title, date and venue. For the first attempt of the name language identification only personal names have been considered and processed in isolation from other information contained in the records. We run our experiments on the DBLP release from February 2008 [5] which has listed 609411 personal names. To increase the accuracy of classification the system only deals with the names whose complete length is ≥ 4, which has amounted to 608350 names.

While the system has shown promissing results during the test runs, applying it to the DBLP brings out a number of differences between the settings:

– Language scope. Presumably DBLP contains names from much more languages than our system in its current state can handle (all Eastern-European, Indian, Korean, Arabic, etc.). To detect such names and avoid them from being randomly assigned to one of the existing categories, we adopt the following method:
  Recall that the weight of a name in the language is determined by the frequency of its n-grams in that language. Hence, names from unknown languages are especially prone to penalties according to the "penalization policy" described in 2.1. Should the name receive at least one penalty in all 14 languages, it is labeled "other" and is sent to the file which collects names from languages not covered by the system.

[4] For the purpose of this work the conditional tetragram metric described above is used.

[5] The up-to-date versions of DBLP are available for download from `http://dblp.uni-trier.de/xml/` in xml format.

– Uncertain names. Even for the 14 languages the system deals with, the decision is not always unambiguous. This holds for the names of closely related languages, for example Portuguese and Spanish, Dutch and German, Danish and Norvegian, etc., because of the overlap in n-grams and similarity of their respective frequencies. Another reason for uncertainty are names whose components are typical for more than one language. For instance "Robert" or "Charles" occur (and are written in the same way) in both, English and French, and assignment of the name "Charles Robert" to English or French is almost equally likely. In terms of the name scores, such cases would have a very small difference between the 1st and 2nd best choices, and thus the classification cannot be accepted with confidence[6]. Such names are assigned the language where they have gained the highest score, but labeled "uncertain".
– Mixed names. Mixed names are the ones, whose components belong to the different languages. For instance, in the name "Thomas Xavier Vintimilla" the first given name is English (Welsh origin), the second one – Spanish (Basque origin, written as Javier in modern Spanish, also popular in France, US), and the family name is probably Spanish. Mixed names do not necessarily have close 1st and 2nd best ranks, and hence are not always recognized as "uncertain". They are often misclassified.

To increase the system's performance in the real life conditions we enhance the model with the co-author network. The idea is that the collaboration between researchers speaking the same language is more widespread than the cross–linguistic one. Thus, if a person whose name is labeled "uncertain" with the highest rank in Italian, has mainly Italian co-authors (as classified by the system), it can be identified as Italian with increased certainty. In the same spirit, misclassified names can be reassigned the most appropriate language category. Of course, this method is not a substitute for the languages that are not covered by our system. However it may help to correct the initial classification by transferring names erroneously labeled "other" to one of the languages known to the system based on the co-author list assignment. On the other hand, co-author classification serves as support for the author name classification, in case they agree.

Bellow we describe the application of co-author network to the personal name language classification in more details.

3 Language Detection Using Co-author Network

3.1 DBLP as a Co-author Network

To conduct the experiments we transform the DBLP into a network of co-authors represented by a graph $G = (V, E)$, where V is the set of vertices which correspond to personal names, and E is the set of edges which are defined by the

[6] In the experiments described here decision is confirmed if the difference between the two highest scores ≥ 0.5.

co-authorship: there is an edge between two authors $\langle a, b\rangle$ if they have at least one common publication. Based on the DBLP data from February 2008, the network graph consists of 609411 vertices, and 3634114 edges. In average, there are 2.51 authors per publication, and 4.1 publications per author, out of which 3.69 are made in collaboration with the other authors. For every co-author b of an author a we calculate the relative strength of their co-authorship via the formula:

$$w_b(a) = \sum_{i=1}^{n} 1/(A_i - 1),$$

where A_i is the number of co-authors in the ith common publication of a and b, and n is the number of the common publications. There are on average 5.96 co-authors per author, and the co-authorship strength across the database is 0.63.

3.2 Language Detection Using Co-author Network

In this enhanced approach the language classification consists of three steps:

- Personal name language detection for every vertex in V. This step is done according to the procedure described in the Subsection 2.1. The result is partitioning of the DBLP personal names into language categories, as described in the Subsection 2.2.
- Verification of the initial classification. The objectif is to determine for every $a \in V$ the dominating language category of his/her co-authors.
- Refine the classification by merging the results of the two independent classifications (via linguistic structure of the name and via co-author network).

We implement three different methods of computing the language category of the co-authors.

3.3 Classification Using Probabilistic Voting Approach

This method represents a kind of "voting system", where each co-author votes for the language to which his personal name has been attributed by the first round of the classification process.[7] We will also describe other more refined models later in this section. Consider the following example: Suppose that out of 30 co-authors the highest vote for a single language (say, Italian) is 10. Is this a chance event or a strong bias towards Italian? In order to determine the threshold we propose the probabilistic method described below.

This method determines how much the probability of selecting one of the 14 languages by co-author voting is higher than a chance selection. We iterate over the co-authors b_i of $a \in V$, count for each language the number of co-authors that have been assigned to it, and determine the language with the largest counter

[7] We consider all $a \in V$ that have ≥ 5 co-authors, and ≥ 3 works produced in collaboration, i.e. sufficient co-authorship strength. In total 131989 DBLP authors pass this criteria.

c_{max}. We assume that the language counters are binomially distributed $B(n,p)$ with $p = 1/14$ (independent choice of one of the 14 languages) and n – being the number of co-authors of a. For some language the probability that the number X of co-authors assigned to it is $< c_{max}$ is expressed by:

$$P(X < c_{max}) = F(c_{max}; n, p),$$

where F is the cumulative function of the binomial distribution. This cumulative function can be evaluated using the regularized incomplete beta function, as follows:

$$F(c_{max}; n, p) = P(X < c_{max}) = I_{1-p}(n - c_{max}, c_{max} + 1),$$

provided that $0 < c_{max} \leq n$. Thus taking into account the 14 languages treated by the system we can compute the probability P that in some language the number of co-authors is higher than c_{max}, applying the formula:

$$P = 1 - I_{1-p}(n - c_{max}, c_{max} + 1)^{14}.$$

If $P < p_{min}$ we accept that having c_{max} co-authors voting for the same language is not a chance event. In our experiments p_{min} is set to 0.01. (We have checked other possibilities for p_{min}, from 0.02 to 0.05, and kept 0.01 as producing the most accurate results).

This model can be further refined due to the following observations:

- By using only a single vote per co-author we loose the possibly relevant infomation that is contained in the second best, third best, etc. languages proposed by our linguistic model. We can still accomodate this information by giving points to the top five languages for each person, with some decay factor. For example: a vote of 1 for the first language, a vote of 0.5 for the second, 0.25 for the third, etc. (decay factor 1/2). The reason why we work with points rather than with linguistic weights is that the later depend on the corpus size, frequency and the total number of the unique n-grams in the language. They are also influenced by the corpus frequency of names. Thus it makes no sense to compare absolute weight values across the languages.
- The second observation is that a co-authorship strength $w_b(a)$ varies between the co-authors and thus giving all the co-authors the same voting power may not be optimal. We may thus weight the vote of each co-author by his/her co-authorship strength with the target author a.

In all these methods we do not consider co-author names labeled "other" because they mainly belong to the languages not covered by the system. If all the co-authors of a given author are "others", the author is skipped.

Finally, we check whether the language category suggested by the co-authors corresponds to the one obtained by the author in the 1st classification step. Results produced by this method are discussed in the following section.

4 Evaluation of the Results and Discussion

We apply the three methods described in Subsection 3.3 to the 131989 DBLP authors who satisfy the co-authorship strength criteria. From that list 100 names have randomly been chosen to assess the quality of the classification. Table 1 summarizes the results.

Table 1. Evaluation of the name language classification using co-author network

Category	True	False	Chance
Methods agree	36	0	–
Methods differ	37	5	–
Chance	–	–	22
Total	73	5	22

We notice that 22 names out of 100 have not been classified because the language selection made by the co-author voting have been considered a chance selection by all the three methods. In the other 36 cases the language selected by the co-authors corresponds to the one initially attributed to the name by the linguistic method, and in 42 cases – the two classifications disagree. To check the correctness of these results we have searched for the information concerning the author's current or past affiliation. As the evaluation table suggests, the co-author based classification is true in most of the cases (only 5 errors out of 78 cases, i.e. above 90% success rate). The match between the linguistic and the co-author based classifications speaks for the hypothesis that people tend to collaborate within monolingual communities. The disagreement between the two usually occurs in one of the following scenarios:

- The name is classified with uncertainty or misclassified. For example in our test set there are 27 such names out of 42, and 7 among them are initially labeled "other" while they actually fall into the scope of languages processed by the system. Due to the co-author based classification we could correct the initial assignment.
- Person works outside of his/her native linguistic environment (for example, in another country). We have encountered 10 such names out of 42 in our test set. In that case co-authors attribute the name to the language of community to which he/she contributes.

The technique-wise comparison shows that all the three methods usually produce the same language selection for a single author. However the method which takes into account the co-authorship strength $w_b(a)$ may select the language of the strongest co-author, if there is one. This feature makes it useful for discoverig special patterns in co-authorship, for example: $\langle professor, PhD - student \rangle$.

5 Summary

In this paper we have described an approach for the personal name language classification using the co-author network. We have developed a voting model of

the language selection and proposed three statistical metrics for calculation of how much the probability of selecting one of the languages by the co-author is higher than a chance selection. We have tested our model on co-author graph built from the DBLP data. The results have shown that the extension of the language classification process with the co-author network may help to improve the initial, based on the author names only, linguistic classification. It may also lead to the discovery of dependencies between the elements of the co-author network, or participation of authors in scientific communitites. Our evaluation has shows that the co-author based identification is correct in more than 90% of the cases.

Acknowledgement

This work has been done in the MINE group/ILIAS laboratory of the Faculty of Computer Science of the University of Luxembourg. I would like to thank my Ph.D. supervisor, Prof. Christoph Schommer, for his helpful advice, valuable comments and fruitful discussions. I would also like to thank Dr. Michael Ley for providing me with the DBLP parsing software.

References

1. Börner, K., Dall'Asta, L., Ke, W., Vespignani, A.: Studying the emerging global brain: Analyzing and visualizing the impact of co-authorship teams. Complexity 10(4) (2005)
2. Google: Google Scholar, http://scholar.google.com/
3. Han, H., Zha, H., Lee, G.C.: Name disambiguation in author citations using a K-way spectral clustering method. In: ACM/IEEE Joint Conference on Digital Libraries, pp. 334–343 (2005)
4. Hiemstra, D., Hauff, C., Jong, F., Kraaij, W.: SIGIR's 30th anniversary: an analysis of trends in IR research and the topology of its community. In: SIGIR Forum, vol. 41(2) (2007)
5. Huang, T., Huang, M.L.: Analysis and Visualization of Co-authorship Networks for Understanding Academic Collaboration and Knowledge Domain of Individual Researchers. In: 4th International Conference on Computer Graphics, Imaging and Visualization, pp. 18–23 (2006)
6. Ke, W., Börner, K., Viswanath, L.: Major Information Visualization Authors, Papers and Topics in the ACM Library. In: 10th IEEE Symposium on Information Visualization (2004)
7. Klink, S., Reuther, P., Weber, A., Walter, B., Ley, M.: Analysing Social Networks Within Bibliographical Data. In: Database and Expert Systems Applications, pp. 234–243 (2006)
8. Lee, C.G., Bollacker, K.D., Lawrence, S.: CiteSeer: An Automatic Citation Indexing System. In: ACM Digital Libraries, pp. 89–98 (1998), http://citeseer.ist.psu.edu/
9. Lee, D., On, B., Kang, J., Park, S.: Effective and scalable solutions for mixed and split citation problems in digital libraries. In: Workshop on Information Quality in Information Systems, pp. 69–76 (2005)

10. Ley, M.: DBLP, Computer Science Bibliography, http://www.informatik.uni-trier.de/~ley/db/
11. Mei, Q., Cai, D., Zhang, D., Zhai, C.: Topic modeling with network regularization. In: 17th International World Wide Web Conferences, pp. 101–110 (2008)
12. Murray, C., Ke, W., Börner, K.: Mapping Scientific Disciplines and Author Expertise Based on Personal Bibliography Files. In: 10th International Conference on Information Visualisation, vol. IV, pp. 258–263 (2006)
13. Reuther, P., Walter, B., Ley, M., Weber, A., Klink, S.: Managing the Quality of Person Names in DBLP. In: European Conference on Digital Libraries, pp. 508–511 (2006)
14. Rodrigues, J.F., Tong, H., Traina, A., Faloutsos, C., Leskovec, J.: GMine: A System for Scalable, Interactive Graph Visualization and Mining. In: 32nd Very Large Data Bases, pp. 1195–1198 (2006)
15. Shannon, C.E.: The Mathematical Theory of Communication. Bell System Technical Journal 27 (1948)
16. SIGIR – Special Interest Group on Information Retrieval, http://www.sigir.org/
17. Smeaton, A.F., Keogh, G., Gurrin, C., McDonald, K., Sødring, T.: Analysis of papers from twenty-five years of SIGIR conferences: what have we been doing for the last quarter of a century? SIGIR Forum 37(1) (2003)
18. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.L.: Probabilistic author-topic models for information discovery. In: Knowledge Discovery and Data Mining, pp. 306–315 (2004)
19. Wikipedia, the Free Encyclopedia, Lists of People, http://en.wikipedia.org/wiki/
20. Zhang, H., Qiu, B., Lee, C.G., Foley, H.C., Yen, J.: An LDA-based Community Structure Discovery Approach for LLarge-Scale Social Networks. In: IEEE International Conference on Intelligence and Security Informatics, pp. 200–207. IEEE Press, New York (2007)
21. Zhang, H., Qiu, B., Lee, C.G., Foley, H.C., Yen, J.: Probabilistic Community Discovery Using Hierarchical Latent Gaussian Mixture Model. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, pp. 663–668 (2007)
22. Detection of Personal Name Language Origin and Applications (submitted, 2008)

On Clustering the Criteria in an Outranking Based Decision Aid Approach

Raymond Bisdorff

University of Luxembourg, FSTC/CSC, 6–rue Richard Coudenhove-Kalergi,
L–1359 Luxembourg
raymond.bisdorff@uni.lu

Abstract. In this paper we discuss the clustering of the set of criteria in a multicriteria decision analysis. Our approach is based on a generalisation of Kendall's rank correlation resulting in the definition of a bipolar ordinal correlation index. A factorial decomposition of this index allows to compute the principal inertia planes of the criteria correlations. The same ordinal correlation index, modelling a symmetric bipolar-valued similarity digraph, allows us to compute a criteria clustering from its maximal cliques.

Introduction

The PROMETHEE authors [1,2] consider very accurately that one of the methodological requisites for an appropriate Multicriteria Decision Aid (MCDA) method is the necessity to provide information on the *conflicting nature* of the criteria. The classical Electre methods [3,4] as well as the recent Rubis best choice method [5] do not provide any such information. In this paper we therefore present several tools that, similar in their operational purpose to the PROMETHEE GAIA plane [2], help illustrating concordance and/or discordance of the criteria with respect to the preferential judgments they show on the given set of decision alternatives. The following example will illustrate our discussion all along the paper.

Example 1 (The Ronda choice decision problem). A family, staying during their holidays in Ronda (Andalusia), is planning the next day's activity. The alternatives shown in Table 1 are considered as potential action. The family members agree to measure their preferences with respect to a set of seven criteria such as the time to attend the place (to be minimised), the required physical investment, the expected quality of the food, touristic interest, relaxation, sun fun & more, ... (see Table 2). The common evaluation of the performances of the nine alternatives on all the criteria results in the performance table shown in Table 3. All performances on the qualitative criteria are marked on a same ordinal scale going from 0 (lowest) to 10 (highest). On the quantitative *Distance* criterion (to be minimized), the required travel time to go to and return from the activity is marked in negative minutes. In order to model only effective preferences, an indifference threshold of 1 point and a preference threshold of 2 points is put

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 409–418, 2008.

Table 1. Ronda example: The set of alternatives

Identifier	Name	Comment
ant	Antequerra	An afternoon excursion to Antequerra and surroundings.
ard	Ardales	An afternoon excursion to Ardales and El Chorro.
be	beach	Sun, fun and more.
crd	Cordoba	A whole day visit by car to Cordoba.
dn	fa niente	Doing nothing.
lw	long walk	A whole day hiking.
mal	Malaga	A whole day visit by car to Malaga.
sev	Sevilla	A whole day visit by car to Sevilla.
sw	short walk	Less than a half day hiking.

Table 2. Ronda example: The set of criteria

Identifier	Name	Comment
cult	*Cultural Interest*	Andalusian heritage.
dis	*Distance*	Minutes by car to go to and come back from the activity.
food	*Food*	Quality of the expected food opportunities.
sun	*Sun, Fun, & more*	No comment.
phy	*Physical Investment*	Contribution to physical health care.
rel	*Relaxation*	Anti-stress support.
tour	*Tourist Attraction*	How many stars in the guide ?

Table 3. Ronda example: The performance table

Criteria	ant	ard	be	crd	dn	lw	mal	sev	sw
cult	7.0	3.0	0.0	10.0	0.0	0.0	5.0	10.0	0.0
dis	-120.0	-100.0	-30.0	-360.0	0.0	-90.0	-240.0	-240.0	0.0
phy	3.0	7.0	0.0	5.0	0.0	10.0	5.0	5.0	5.0
rel	1.0	5.0	8.0	3.0	10.0	5.0	3.0	3.0	6.0
food	8.0	10.0	4.0	8.0	10.0	1.0	8.0	10.0	1.0
sun	0.0	3.0	10.0	3.0	1.0	3.0	8.0	5.0	5.0
tour	5.0	7.0	3.0	10.0	0.0	8.0	10.0	10.0	5.0

on the qualitative performance measures. On the *Distance* criterion, an indifference threshold of 20 min, and a preference threshold of 45 min. is considered. Furthermore, a difference of more than two hours to attend the activity's place is considered to raise a veto (see Table 4).

The individual criteria each reflect one or the other member's preferential point of view. Therefore they are judged equi-significant for the best action to be eventually chosen.

How do the criteria express their preferential view point on the set of alternatives? For instance the *Tourist Attraction* criterion appears to be in its preferential judgments somehow positively correlated with both the *Cultural Interest* and the *Food* criteria. It is also apparent that the *Distance* criterion is somehow negatively correlated to these latter criteria. How can we explore and illustrate these intuitions?

In a given MCDA, where a certain set of criteria is used for solving a given decision problem, it is generally worthwhile analysing to what extent the

Table 4. Ronda example: Preference discrimination thresholds

Criterion	Thresholds indifference	preference	veto
cult	1pt	2pts	-
dis	20min.	45min.	121min.
food	1pt	2pts	-
sun	1pt	2pts	-
phy	1pt	2pts	-
rel	1pt	2pts	-
tour	1pt	2pts	-

criteria vary in their relational judgments concerning the pairwise comparison of performances of the alternatives. Illustrating such similarities and dissimilarities between criteria judgments is indeed the very purpose of this paper. First, we present a bipolar-valued ordinal criteria correlation index, generalising Kendall's τ [6], and illustrating the preferential distance between the criterial judgments. In a second section we show how to decompose this correlation index into its principal components. In a third section, following an earlier work of ours [11], we propose a credibility level indexed clustering of the criteria based on the extraction of maximal bipolar-valued cliques observed in the associated criteria similarity digraph.

1 A Bipolar-Valued Ordinal Criteria Correlation Index

Let us introduce our notations. We consider a finite set A of n alternatives and denote by x and y any two alternatives. We consider also a set F of outranking criteria [4] denoted by variables i or j, with $k = 0, 1, ...$ discrimination thresholds. The performance of an alternative x on criterion i is denoted by x_i.

Example 2. The four discrimination thresholds we may observe on each criterion i for instance in the Rubis choice method [5] are: – "weak preference"[1] wp_i $(0 < wp_i)$, – "preference" p_i $(wp_i \leqslant p_i)$, – "weak veto" wv_i $(p_i < wv_i)$, and – "veto" v_i $(wv_i \leqslant v_i)$. Each difference $(x_i - y_i)$ may thus be classified into one and only one of the following nine cases:

$(\ggg)$	"*veto against* $x \leqslant y$"	$\Leftrightarrow$	$v_i \leqslant (x_i - y_i)$
$(\gg)$	"*weak veto against* $x \leqslant y$"	$\Leftrightarrow$	$wv_i \leqslant (x_i - y_i) < v_i$
$(>)$	"x *better than* y"	$\Leftrightarrow$	$p \leqslant (x_i - y_i)$
$(\geqslant)$	"x *better than or equal* y"	$\Leftrightarrow$	$wp_i \leqslant (x_i - y_i) < p_i$
$(=)$	"x *indifferent to* y"	$\Leftrightarrow$	$-wp_i < (x_i - y_i) < wp_i$
$(\leqslant)$	"x *worse than or indifferent to* y"	$\Leftrightarrow$	$-p_i < (x_i - y_i) \leqslant -wp_i$
$(<)$	"x *worse than* y"	$\Leftrightarrow$	$-wp_i < (x_i - y_i) \leqslant -p_i$
$(\ll)$	"*weak veto against* $x \geqslant y$"	$\Leftrightarrow$	$-v_i < (x_i - y_i) \leqslant -wp_i$
$(\lll)$	"*veto against* $x \geqslant y$"	$\Leftrightarrow$	$(x_i - y_i) \leqslant -v_i$

[1] In some cases it may be useful to replace the weak preference threshold, defining an open indifference interval on the criterion scale, with an indifference threshold $0 \leqslant h$ defining a closed indifference interval and leaving open the weak preference interval (see [5]).

In general, let us consider on each criterion i, supporting a set of discrimination thresholds p_r $(r = 1, .., k)$ such that $0 < p_1 \leqslant ... \leqslant p_k$, the Kendall vector (see [7]) gathering the classification of all possible differences $(x_i - y_i)$ into one of the following $2k + 1$ cases:

$$(x_i - y_i) \in \begin{cases} (>_k) & \text{if} \quad p_k \leqslant (x_i - y_i) \\ (>_r) & \text{if} \quad p_r \leqslant (x_i - y_i) < p_{r+1}, \text{for } r = 1, ...k-1 \\ (=) & \text{if} \quad -p_1 < (x_i - y_i) < p_1 \\ (<_r) & \text{if} \quad -p_{r+1} < (x_i - y_i) \leqslant -p_r, \text{for } r = 1, ...k-1 \\ (<_k) & \text{if} \quad (x_i - y_i) \leqslant -p_k \end{cases} \tag{1}$$

Comparing the preferential view point of two criteria i and j, we say that x and y are *concordantly* (resp. *discordantly*) compared if $(x_i - y_i)$ and $(x_j - y_j)$ are classified into the same category (resp. different categories) on both criteria. This is the case if position (i, j) in both Kendall vectors is of the same (resp. different) value. There are $n(n-1)$ distinct ordered pairs of performances and each pair (x, y) is thus either concordantly or discordantly classified. Please notice that we may well compare two criteria with a different number of discrimination thresholds. The only semiotic restriction we require here is that the preferential meanings of the k thresholds are the same for all criteria in the given family F. Denoting by S_{ij} the number c_{ij} of concordantly classified minus the number d_{ij} of discordantly classified ordered pairs, the *ordinal criteria correlation index* $\widetilde{T}$ is defined on $F \times F$ as

$$\tilde{T}(i, j) = \frac{c_{ij} - d_{ij}}{c_{ij} + d_{ij}} = \frac{S_{ij}}{n(n-1)}. \tag{2}$$

Property 1. The ordinal criteria correlation index $\tilde{T}$ is symmetrically valued in the rational bipolar credibility domain $[-1, 1]$ (see [8,5]).

Proof. If all pairs of alternatives are concordantly (discordantly) classified by both criteria, $d_{ij} = 0$ (resp. $c_{ij} = 0$) and $\widetilde{T}(i, j) = 1.0$ (resp. -1.0). If $\widetilde{T}(i, j) > 0$ (resp. < 0) both criteria are more *similar than dissimilar* (resp. *dissimilar than similar*) in their preferential judgments. When $\widetilde{T}(i, j) = 0.0$, no conclusion can be drawn. The linear structure of the criterion scale and the relational coherence of the discrimination thresholds imply that a performance difference $(x_i - y_i)$ is classified in one and only one case. Furthermore, the case of $(x_i - y_i)$ corresponds bijectively to a unique symmetric case classifying the reversed difference $(y_i - x_i)$. Hence, the pair (x, y) is concordantly classified by criteria i and j if and only if the symmetric pair (y, x) is concordantly classified by the same two criteria. □

Property 2. If i and j are two perfectly discriminating criteria, i.e. they admit a single preference threshold $p_1 = \epsilon$, and we don't observe ties in the performance table then $\widetilde{T}(i, j)$ is identical with the classical τ of Kendall [6].

Proof. In this case, both the Kendall vectors of criteria i and j contain only the two possible cases: – case $(>_1)$: $(x_i - y_i) \geqslant \epsilon$, and – case $(<_1)$: $(x_i - y_i) \leqslant \epsilon$. Denoting

by p_{ij} the number of pairs (x, y) in $A \times A$ such that conjointly $(x_i - y_i) \geqslant \epsilon$ and $(x_j - y_j) \geqslant \epsilon$ we obtain indeed

$$\widetilde{T}(i,j) = (2 \times \frac{2p_{ij}}{n(n-1)}) - 1, \quad \forall (i,j) \in F \times F, \tag{3}$$

i.e. Kendall's original τ definition (see [6]). □

It is worthwhile noticing that the classical problem for applying Kendall's τ to a situation with ties is here coherently resolved. Indeed, Equation 2 generalises Kendall's rank correlation index to any family of homogeneous semiorders (see [9] Chapter 3).

Example 3 (The Ronda decision problem – continued). Computing our ordinal criteria correlation index $\widetilde{T}$ (see Equation 2) on the set of seven criteria we obtain the results shown in Table 5. As initially suspected, on the one hand, we observe here that the performances on the criteria *Cultural interest* and *Tourist Attraction*, and *Physical Investment* lead to positively correlated preferential judgments ($\widetilde{T}(\text{cult}, \text{tour}) = +0.28$ and $\widetilde{T}(\text{phy}, \text{tour}) = +0.33$). On the other hand, the performances observed on criteria *Distance* and *Cultural Interest* or *Tourist Attraction* lead to nearly completely opposed preferential statements ($\widetilde{T}(\text{dis}, \text{tour}) = -0.92$ and $\widetilde{T}(\text{dis}, \text{cult}) = -0.89$).

As these couples of concordant and/or discordant criteria play an essential role in the actual difficulty of the decision making process, we look for a systematic graphical illustration of the ordinal criteria correlation index.

Table 5. Ronda example: The ordinal criteria correlation table

$\widetilde{T}$	dis	phy	rel	food	sun	tour
cult	−0.89	−0.17	−0.81	+0.00	−0.39	+0.28
dis		−0.72	−0.08	−0.67	−0.39	−0.92
phy			−0.17	−0.39	−0.28	+0.33
rel				−0.25	−0.17	−0.53
food					−0.56	−0.17
sun						−0.03

2 Principal Component Analysis of the Criteria Correlation

A most suitable tool is given by the classical Principal Component Analysis – PCA [10]. We may uncover the principal components of $\widetilde{T}$ by computing the eigen-vectors of its associated covariance. Projecting the criteria points in the covariance eigen-space along the principal coordinates explaining the largest part of the total variance reveals the major agreements and oppositions between the preferential judgments as expressed by the criteria on the given set of alternatives.

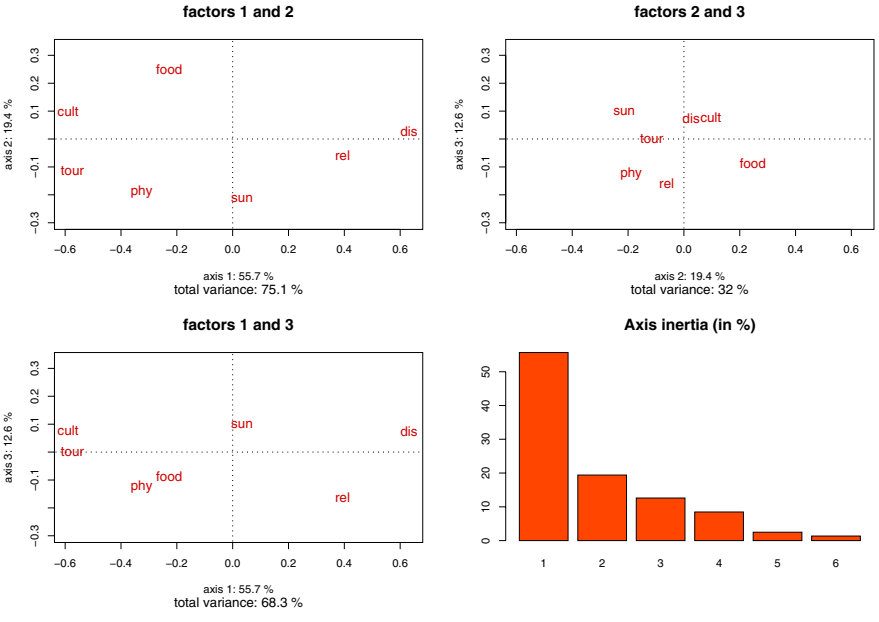


Fig. 1. Ronda example: Results of the PCA

Example 4 (The Ronda decision problem – continued). Such PCA results, computed from the $\widetilde{T}$ index observed in the Ronda example, are shown in Figure 1. As expected, the first and largely prominent opposition – gathering 55.7% of the total variance – is observed between, on the one hand, both criteria *Cultural Interest* and *Tourist Attraction*, and, on the other hand, criterion *Distance* and, to a lesser extent, criterion *Relaxation*. The second factorial axis – already much less prominent (only 19.5% of total variance) – shows an opposition between, on the one hand, the *Food* criterion and, on the other hand, both the *Sun, Fun & more* and the *Physical Investment* criteria. It is furthermore worthwhile noticing that all seven criteria appear in a more or less elliptic layout in the main principal plane (gathering 75.1% of all variance) and thereby indicate that each one owns a specific preferential judgment behaviour, somehow different from all the others.

The PCA of $\widetilde{T}$ is much like the well known PROMETHEE Gaia approach [2]. Main difference is that the Gaia PCA is realized on the covariance of the rows – describing the alternatives – of the single net flows matrix (see [2]). Recall that the single net flow for alternative x on criteria i is the normalized difference between the number of times x is preferred to the other alternatives minus the number of times the other alternatives are preferred to x. The Gaia plane therefore shows the projection of the alternatives in the plane of the two most prominent principal axes. The criteria are there only indirectly represented as supplementary points, the unit vectors of the coordinate axis representing each

criteria. Practical experiments have shown that very similar result to ours would however appear when realizing a PCA on the covariance, not of the rows, but of the columns – describing the criteria – of the single net flows matrix. Main advantage of our $\widetilde{T}$ measure is, nonetheless, that the distance between the criteria's preferential judgments is not computed from a compound preference situation, but takes into account the indifference situation as well as all k discriminated preference levels a criterion may, the case given, attach to a given performance difference on all pairs of alternatives.

If the PCA of the criteria correlation index $\widetilde{T}$ reveals very convincingly the most prominently opposed criteria, the projection of the criteria into the main principal planes also illustrates quite well the potential proximities between criteria (see the position of criteria *Cultural Interest* and *Tourist Attraction* for instance in Figure 1). In order to qualify the credibility of such proximities, we finally propose a bipolar-valued clustering based again on the ordinal criteria correlation index $\widetilde{T}$.

3 Bipolar-Valued Clustering from the Criteria Correlation Index $\widetilde{T}$

For this last approach, we make use of Property 1 which tells us that the index $\widetilde{T}$ represents a bipolar-valued characteristic denotation of the propositional statement "*criteria i and j express similar preferential statements on A*". We consider indeed this statement to be more or less validated if both criteria are concordant on a *majority* of pairwise comparisons and discordant on a minority ones. In this sense, $\widetilde{T}$ is characterising a bipolar-valued *similarity* graph, we denote by $\widetilde{S}(F, \widetilde{T})$ or $\widetilde{S}$ for short. Following from the logical denotation of the bipolar valuation, we say that there is an arc between i and j if $\widetilde{T}(i,j) > 0$ (see [8]). Similarly, a clique C in $\widetilde{S}$ is a subset of criteria such that for all i and j in C, we have $\widetilde{T}(i,j) \geqslant 0$. [2]

In general, we may associate a crisp graph $S(F,T)$ with $\widetilde{S}$, where $T = \{(i,j)|\widetilde{T}(i,) > 0\}$. All properties of S are canonically transferred to $\widetilde{S}$. For instance, S is a symmetric digraph (see Property 1), so is $\widetilde{S}$.

Example 5 (The Ronda decision problem – continued). The criteria similarity graph in the Ronda example contains only three edges: – between *Physical Investment* and *Tourist Attraction* ($\widetilde{T}$(phy,tour) = 0.33), – between *Tourist Attraction* and *Cultural Interest* ($\widetilde{T}$(tour,cult) = 0.28), and – the weak (or potential) similarity between criteria *Food* and *Cultural Interest* ($\widetilde{T}$(food,cult) = 0.0). Notice that the similarity relation is not transitive (a fact easily explainable from Figure 1).

[2] We admit here a weak notion of a bipolar-valued clique by including possibly indeterminate similarity situations. A strict bipolar-valued clique concept would require a strictly positive valuation.

What we are looking for are maximal cliques, i.e. subsets C of criteria which verify both the following properties:

1. *Internal stability*: all criteria in C are similar, i.e. the subgraph $(C, \widetilde{T}_{|C})$ is a clique;
2. *External stability*: if a criteria i is not in C, there must exist a criteria j in C such that $\widetilde{T}(i,j) < 0$ and $\widetilde{T}(j,i) < 0$.

For any $C \in F$, we denote by $\Delta^{int}(C)$ (resp. $\Delta^{ext}(C)$) its credibility of being internally (resp. externally) stable:

$$\Delta^{int}(C) = \begin{cases} 1.0 & \text{if} \quad |C| = 1, \\ \min_{i \in C} \min_{j \in C}^{j \neq i} \left(\widetilde{T}(i,j)\right) & \text{otherwise.} \end{cases} \tag{4}$$

$$\Delta^{ext}(C) = \begin{cases} 1.0 & \text{if} \quad C = F, \\ \min_{i \in F}^{i \notin C} \max_{j \in C} \left(-\widetilde{T}(i,j)\right) & \text{otherwise.} \end{cases} \tag{5}$$

Property 3. A subset C of criteria is a maximal clique of the similarity graph $\widetilde{S} \equiv (F, \widetilde{T})$ if and only if both $\Delta^{int}(C) \geqslant 0$ and $\Delta^{ext}(C) > 0$.

Proof. Condition $\Delta^{int}(C) \geqslant 0$ directly implies that $(C, \widetilde{T}_{|C})$ is a clique and condition $\Delta^{ext}(C) > 0$ implies that, for any criterion i not in C, there exists at least one criterion j in C such that $\widetilde{T}(i,j) < 0$. □

Computing maximal cliques in a graph is equivalent to the problem of computing maximal independent sets in the dual graph. These problems are in theory algorithmically difficult [12]. Considering however the very low dimension of the set of criteria in a common MCDA problem, there is no operational difficulty here for the decision aid practice. The credibility level $\min(\Delta^{ext}, \Delta^{int})$ of the resulting maximal cliques may eventually lead to a bipolar-valued clustering of the family of criteria (see [11]).

Example 6 (The Ronda decision problem – continued). The clustering results are shown in Table 6.

The most validated maximal cliques (at credibility level 58.33%[3]) are the pairs (*Physical Investment*, *Tourist Attraction*) and (*Tourist Attraction*, *Cultural Interest*). At level 54.17%, both the criteria *Distance* and *Relaxation* are singleton maximal cliques, followed at level 51.39% by the criterion *Sun, Fun & more*. Finally, a potential maximal clique is the pair (*Cultural Interest*, *Food*). The credibility level indexed clustering results are shown in Figure 2.

With non-redundant and preferentially independent criteria, we may expect in general very small maximal cliques and singletons. Monte Carlo experiments with random performance tableaux confirm indeed this sparsity of the criteria clustering in normal MCDA problems.

[3] The credibility levels are expressed as $(\min(\Delta^{ext}, \Delta^{int})+1.0)/2.0$ in the $[0,1]$ interval.

Table 6. Ronda example: Clustering the criteria

Maximal cliques	credibility level (in%^{3})	stability external	stability internal
{phy,tour}	58.33	+0.167	+0.333
{tour,cult}	58.33	+0.167	+0.278
{dis}	54.17	+0.083	+1.00
{rel}	54.17	+0.083	+1.00
{sun}	51.39	+0.028	+1.00
{cult,food}	50.00	+0.167	0.0

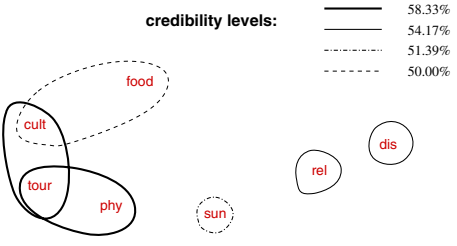


Fig. 2. Ronda example: The bipolar-valued criteria clusters

Conclusion

Despite the obvious importance of the methodological requisite for a suitable MCDA approach to offer tools for illustrating preferential agreements and/or oppositions between the criteria, no specific formal methodological contribution, apart from the PROMETHEE Gaia plane, has been made in the general context of the outranking based MCDA methods. This paper fills this gap with a generalisation of Kendall's rank correlation τ measure to the pairwise comparison of the preferential judgements the criteria apply to a given set of alternatives. This new ordinal criteria correlation index may be used, on the one hand, for graphically illustrating oppositions and agreements between criteria with the help of a PCA similar to the Gaia approach. On the other hand, the same ordinal correlation index may also be used for extracting in a decreasing level of credibility the maximal cliques from a bipolar-valued criteria similarity graph.

References

1. Brans, J.P.: L'ingénierie de la dcision: Élaboration d'instruments d'aide à la décision. La méthode PROMETHEE. In: Nadeau, R., Landry, M. (eds.) L'aide à la décision: Nature, Instruments et Perspectives d'avenir, pp. 183–213. Presses de l'Université Laval, Québec (1982)
2. Brans, J.P., Mareschal, B.: The Promcalc and Gaia decision-support system for multicriteria decision aid. Dec. Sup. Sys. 12, 297–310 (1994)

3. Roy, B.: Méthodologie Multicritère d'Aide à la Décision. Economica Paris (1985)
4. Roy, B., Bouyssou, D.: Aide Multicritére à la Décision: Méthodes et Cas. Economica Paris (1993)
5. Bidorff, R., Meyer, P., Roubens, M.: RUBIS: a bipolar-valued outranking method for the choice problem. 4OR, pp. 1–27. Springer, Heidelberg (2008)
6. Kendall, M.G.: Rank Correlation Methods. Hafner Publishing Co., New York (1955)
7. Degenne, A.: Techniques ordinales en analyse des données statistique. Hachette, Paris (1972)
8. Bisdorff, R.: Logical Foundation of Multicriteria Preference Aggregation. In: Bouyssou, D., et al. (eds.) Essay in Aiding Decisions with Multiple Criteria, pp. 379–403. Kluwer Academic Publishers, Dordrecht (2002)
9. Bouyssou, D., Marchant, Th., Pirlot, M., Tsoukias, A., Vincke, Ph.: Evaluation and Decision Models with Multiple Criteria. Springer, New York (2006)
10. Benzecri, J.P.: L'analyse des données – Tome 2: L'Analyse des correspondances. Dunod Paris (1976)
11. Bidorff, R.: Electre-like clustering from a pairwise fuzzy proximity index. Eur. Jour. Oper. Res. 138, 320–331 (2002)
12. Bisdorff, R.: On enumerating the kernels in a bipolar-valued outranking digraph. Annales du Lamsade 6, 1–38 (2006)

A Fast Parallel SVM Algorithm for Massive Classification Tasks

Thanh-Nghi Do[1], Van-Hoa Nguyen[2], and François Poulet[2]

[1] CIT, CanTho University, VietNam
dtnghi@cit.ctu.edu.vn
[2] IRISA, Rennes, France
{vhnguyen,francois.poulet}@irisa.fr

Abstract. The new parallel incremental Support Vector Machine (SVM) algorithm aims at classifying very large datasets on graphics processing units (GPUs). SVM and kernel related methods have shown to build accurate models but the learning task usually needs a quadratic programming, so that the learning task for large datasets requires big memory capacity and a long time. We extend the recent finite Newton classifier for building a parallel incremental algorithm. The new algorithm uses graphics processors to gain high performance at low cost. Numerical test results on UCI, Delve dataset repositories showed that our parallel incremental algorithm using GPUs is about 45 times faster than a CPU implementation and often significantly over 100 times faster than state-of-the-art algorithms LibSVM, SVM-perf and CB-SVM.

Keywords: Support vector machines, incremental learning, parallel algorithm, graphics processing unit, massive data classification.

1 Introduction

Since SVM learning algorithms were first proposed by Vapnik [26], they have been shown to build accurate models with practical relevance for classification, regression and novelty detection. Successful applications of SVMs have been reported for such varied fields as facial recognition, text categorization and bioinformatics [14]. In particular, SVMs using the idea of kernel substitution have been shown to build good models, and they have become increasingly popular classification tools.

In spite of the prominent properties of SVMs, current SVMs can not easily deal with very large datasets. A standard SVM algorithm requires solving a quadratic program (QP); so its computational cost is at least O(m2), where m is the number of training datapoints. Also, the memory requirements of SVM frequently make it intractable. Unfortunately, real-world databases doubles every 9 months [12], [16]. There is a need to scale up these learning algorithms for dealing with massive datasets. Effective heuristic methods to improve SVM learning time divide the original quadratic program into series of small problems [2], [21], [22]. Incremental learning methods [3], [7], [9], [10], [13], [23], [24] improve memory performance for massive datasets by updating solutions in a growing training

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 419–428, 2008.

set without needing to load the entire dataset into memory at once. Parallel and distributed algorithms [9], [23] improve learning performance for large datasets by dividing the problem into components that execute on large numbers of networked personal computers (PCs). Active learning algorithms [8], [25] choose interesting datapoint subsets (active sets) to construct models, instead of using the whole dataset they can not deal easily with very large datasets.

In this paper, we describe methods to build the incremental and parallel Newton SVM algorithm for classifying very large datasets on GPUs, for example, a Nvidia GeForce 8800 GTX graphics card. Our work is based on Newton SVM classifiers proposed by Mangasarian [17]. He proposed to change the margin maximization formula and add with a least squares 2-norm error to the standard SVM and then this brings out an unconstrained optimization which is solved by the finite stepless Newton method. The Newton SVM formulation requires thus only solutions of linear equations instead of QP. This makes training time very short. We have extended Newton SVM in two ways.

1. We developed an incremental algorithm for classifying massive datasets (billions of datapoints) of dimensionality up to 10^3.

2. Using a GPU (massively parallel computing architecture), we developed a parallel version of incremental Newton SVM algorithm to gain high performance at low cost.

Some performances in terms of learning time and accuracy are evaluated on the UCI repository [1] and Delve [6], including Forest cover type, KDD cup 1999, Adult and Ringnorm datasets. The results showed that our algorithm using GPU is about 45 times faster than a CPU implementation. An example of the effectiveness of the new algorithms is their performance on the 1999 KDD cup dataset. They performed a binary classification of 5 million datapoints in a 41-dimensional input space within 18 seconds on the Nvidia GeForce 8800 GTX graphics card (compared with 552 seconds on a CPU, Intel core 2, 2.6 GHz, 2 GB RAM). We also compared the performances of our algorithm with the highly efficient standard SVM algorithm LibSVM [4] and with two recent algorithms, SVM-perf [15] and CB-SVM [28].

The remainder of this paper is organized as follows. Section 2 introduces Newton SVM classifiers. Section 3 describes how to build the incremental learning algorithm with the Newton SVM algorithm for classifying large datasets on CPUs. Section 4 presents a parallel version of the incremental Newton SVM using GPUs. We present numerical test results in section 5 before the conclusion and future work.

Some notations are used in this paper. All vectors are column vectors unless transposed to row vector by a T superscript. The inner dot product of two vectors, x, y is denoted by $x.y$. The 2-norm of the vector x is denoted by $\|x\|$. The matrix $A[mxn]$ is m datapoints in the n-dimensional real space R^n. e will be the column vector of 1. w, b will be the normal vector and the scalar of the hyperplane. z is the slack variable and C is a positive constant. I denotes the identity matrix.

2 Newton Support Vector Machine

Let us consider a linear binary classification task, as depicted in Figure 1, with m datapoints x_i $(i = 1, \ldots, m)$ in the n-dimensional input space R^n. It is represented by the mxn matrix A, having corresponding labels $y_i = \pm 1$, denoted by the mxm diagonal matrix D of ± 1 (where $D_{i,i} = 1$ if x_i is in class $+1$ and $D_{i,i} = -1$ if x_i is in class -1).

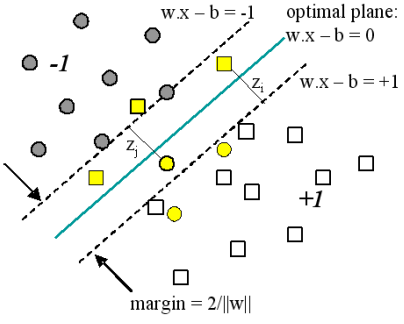


Fig. 1. Linear separation of the datapoints into two classes

For this problem, the SVM algorithm try to find the best separating plane (denoted by the normal vector $w \in R^n$ and the scalar $b \in R^1$), i.e. furthest from both class $+1$ and class -1. It can simply maximize the distance or margin between the supporting planes for each class ($x.w - b = +1$ for class $+1$, $x.w - b = -1$ for class -1). The margin between these supporting planes is $2/\|w\|$ (where $\|w\|$ is the 2-norm of the vector w). Any point x_i falling on the wrong side of its supporting plane is considered to be an error (having corresponding slack value $z_i > 0$). Therefore, a SVM algorithm has to simultaneously maximize the margin and minimize the error. This is accomplished through the following QP (1):

$$\begin{aligned} \min f(w, b, z) &= (1/2)\|w\|^2 + Ce^T z \\ s.t. &: D(Aw - eb) + z \geq e \end{aligned} \tag{1}$$

where $z \in R^m$ is the non negative slack vector and the positive constant $C \in R^1$ are used to tune errors, margin size, respectively.

The plane (w, b) is obtained by solving the QP (1). Then, the classification function of a new datapoint x based on the plane is: $predict(x) = sign(w.x - b)$

SVM can use some other classification functions, for example a polynomial function of degree d, a RBF (Radial Basis Function) or a sigmoid function. To change from a linear to non-linear classifier, one must only substitute a kernel evaluation in (1) instead of the original dot product. More details about SVM and others kernel-based learning methods can be found in [5].

Recent developments for massive linear SVM algorithms proposed by Mangasarian [17], [18] reformulate the classification as an unconstrained optimization. By changing the margin maximization to the minimization of $(1/2)\|w, b\|^2$

and adding with a least squares 2-norm error, the SVM algorithm reformulation with linear kernel is given by the QP (2).

$$\min f(w,b,z) = (1/2)\|w,b\|^2 + (C/2)\|z\|^2 \quad s.t. : D(Aw - eb) + z \geq e \tag{2}$$

where z is the non negative slack vector and the positive constant C are used to tune errors, margin size.

The formulation (2) can be rewritten by substituting for $z = [e - D(Aw - eb)]_+$ (where $(x)_+$ replaces negative components of a vector x by zeros) into the objective function f. We get an unconstrained problem (3):

$$\min f(w,b) = (1/2)\|w,b\|^2 + (C/2)\|[e - D(Aw - eb)]_+\|^2 \tag{3}$$

By setting $[w_1 w_2 \ldots w_n b]^T$ to u and $[A - e]$ to H, then the SVM formulation (3) is rewritten by (4):

$$\min f(w,b) = (1/2)u^T u + (C/2)\|(e - DHu)_+\|^2 \tag{4}$$

Mangasarian [17] has shown that the finite stepless Newton method can be used to solve the strongly convex unconstrained minimization problem (4). The algorithm is described in figure 2. Mangasarian has proved that the sequence u_i of the algorithm terminates at the global minimum solution. In most of the tested cases, the Newton algorithm has given the good solution with a number of iterations varying between 5 and 8.

The SVM formulation (4) requires thus only solutions of linear equations of (w,b) instead of QP. If the dimensional input space is small enough (less than 10^3), even if there are millions datapoints, the Newton SVM algorithm is able to classify them in minutes on a PC.

- Input: training dataset represented by A and D matrices
- Starting with $u_0 \in R^{n+1}$ and $i = 0$
- Repeat
  1) $u_{i+1} = u_i - \partial^2 f(u_i)^{-1} \nabla f(u_i)$
  2) $i = i + 1$

  Until $\nabla f(u_i) = 0$
- Return u_i

Where the gradient of f at u_i,

$$\nabla f(u_i) = C(-DH)^T(e - DHu_i)_+ + u_i \tag{5}$$

and the generalized Hessian of f at u_i,

$$\partial^2 f(u_i) = C(-DH)^T diag([e - DHu_i]_*)(-DH) + I \tag{6}$$

with $diag([e - DHu_i]_*)$ denotes the $(n+1)x(n+1)$ diagonal matrix whose j^{th} diagonal entry is sub-gradient of the step function $(e - DHu_i)_+$

Fig. 2. Newton SVM algorithm

3 Incremental Newton SVM Algorithm

Although the Newton SVM algorithm is fast and efficient to classify large datasets, it needs load whole dataset in the memory. With a large dataset e.g. one billion datapoints in 20 dimensional input, Newton SVM requires more than 80 GB RAM. Any machine learning algorithm has some difficulties to deal with the challenge of large datasets. Our investigation aims at scaling up the Newton SVM algorithm to classify very large datasets on PCs (Intel CPUs). The incremental learning algorithms are a convenient way to handle very large datasets because they avoid loading the whole dataset in main memory: only subsets of the data are considered at any one time and update the solution in growing training set. The main idea is to incrementally compute the gradient of f and the generalized Hessian of f at u for each iteration in the finite Newton algorithm described in figure 3.

Suppose we have a very large dataset decomposed into small blocks by rows A_i, D_i. The incremental algorithm of the Newton SVM can simply incrementally compute the gradient and the generalized Hessian of f by the formulation (5) and (6). Consequently, the incremental Newton SVM algorithm can handle massive datasets on a PC. If the dimension of the input space is small enough (less than 10^3), even if there are billions datapoints, the incremental Newton SVM algorithm is able to classify them on a standard personal computer (Pentium IV, 512 MB RAM). The algorithm only needs to store a small $(n+1)x(n+1)$ matrix and two $(n+1)x1$ vectors in memory between two successives steps (where n is number of dimensions). The accuracy of the incremental algorithm is exactly the same as the original one.

4 Parallel Incremental Newton SVM Using GPUs

The incremental Newton SVM algorithm described above is able to deal with very large datasets on a PC. However it only runs on one single processor. We have extended it to build a parallel version using a GPU.

During the last decade, GPUs described in [27] have developed as highly specialized processors for the acceleration of raster graphics. The GPU has several advantages over CPU architectures for highly parallel, compute intensive workloads, including higher memory bandwidth, significantly higher floating-point, and thousands of hardware thread contexts with hundreds of parallel compute pipelines executing programs in a single instruction multiple data (SIMD) mode. The GPU can be an alternative to CPU clusters in high performance computing environments. Recent GPUs have added programmability and been used for general-purpose computation, i.e. non-graphics computation, including physics simulation, signal processing, computational geometry, database management, computational biology, data mining.

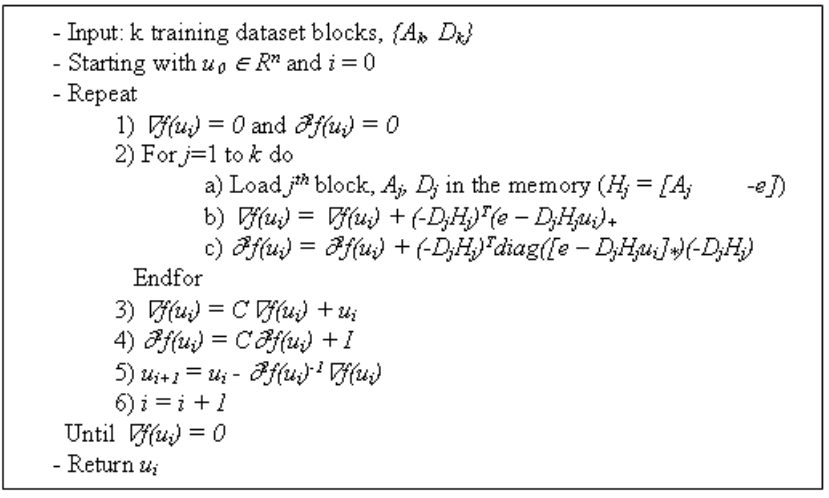


Fig. 3. Incremental Newton SVM algorithm

NVIDIA has introduced a new GPU, i.e. Geforce 8800 GTX and a C-language programming API called CUDA [19] (compute unified device architecture). A block diagram of the NVIDIA Geforce 8800 GTX architecture is comprised of 16 multiprocessors. Each multiprocessor has 8 SPs (streaming processors) for a total of 128 SPs. Each group of 8 SPs shares one L1 data cache. A SP contains a scalar ALU (arithmetic logic unit) and can perform floating point operations. Instructions are executed in a SIMD mode. The NVIDIA Geforce 8800 GTX has 768 MB of graphics memory, with a peak observed performance of 330 GFLOPS and 86 GB/s peak memory bandwidth. This specialized architecture can sufficiently meet the needs of many massively data-parallel computations. In addition, NVIDIA CUDA also provides a C-language API to program the GPU for general purpose applications. In CUDA, the GPU is a device that can execute multiple concurrent threads. The CUDA software stack is composed of a hardware driver, an API, its runtime and higher-level mathematical libraries of common usage, an implementation of Basic Linear Algebra Subprograms (CUBLAS [20]). The CUBLAS library allows access to the computational resources of NVIDIA GPUs. The basic model by which applications use the CUBLAS library is to create matrix and vector objects in GPU memory space, fill them with data, call a sequence of CUBLAS functions, and, finally, upload the results from GPU memory space back to the host. Furthermore, the datatransfer rate between GPU and CPU memory is about 2 GB/s.

Thus, we developed a parallel version of incremental Newton SVM algorithm based on GPUs to gain high performance at low cost. The parallel incremental implementation in figure 4 using the CUBLAS library performs matrix computations on the GPU massively parallel computing architecture. Note that in CUDA/CUBLAS, the GPU can execute multiple concurrent threads. Therefore, parallel computations are done in the implicite way.

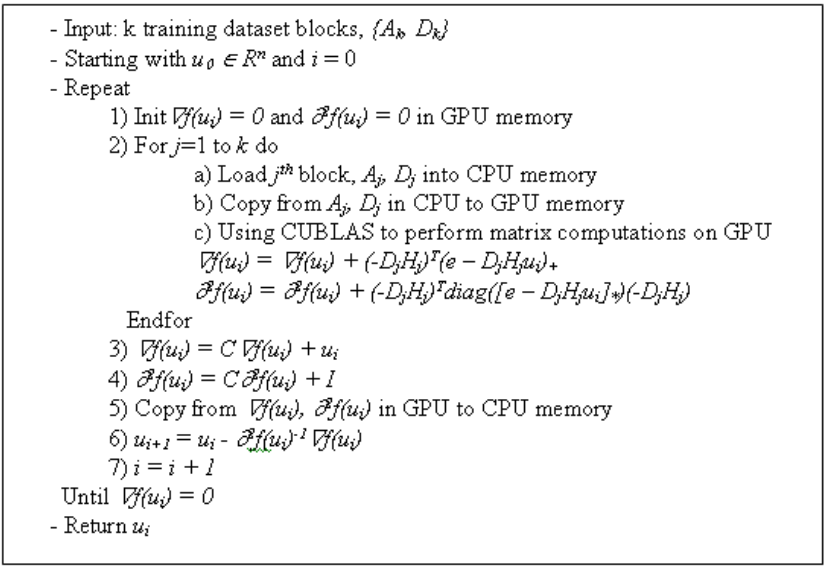


Fig. 4. Parallel incremental Newton SVM algorithm using GPUs

First, we split a large dataset A, D into small blocks of rows A_j, D_j. For each incremental step, a data block A_j, D_j is loaded to the CPU memory; a datatransfer task copies A_j, D_j from CPU to GPU memory; and then GPU computes in the parallel way the sums of: $\nabla f(u_i) = \nabla f(u_i) + (-D_j H_j)^T (e - D_j H_j u_i)_+$ and
$\partial^2 f(u_i) = \partial^2 f(u_i) + (-D_j H_j)^T diag([e - D_j H_j u_i]^*)(-D_j H_j)$
Then the results $\nabla f(u_i)$ and $\partial^2 f(u_i)$ are uploaded from GPUs memory back to CPU memory to update u at the ith iteration. The accuracy of the new algorithm is exactly the same as the original one.

5 Numerical Test Results

We prepared an experiment setup using a PC, Intel Core 2, 2.6 GHz, 2 GB RAM, a Nvidia GeForce 8800 GTX graphics card with NVIDIA driver version 6.14.11.6201 and CUDA 1.1, running Linux Fedora Core 6. We implemented two versions (GPU and CPU code) of incremental Newton SVM algorithm in C/C++ using NVIDIA's CUDA, CUBLAS API [19], [20] and the high performance linear algebra libraries, Lapack++ [11]. The GPU implementation results are compared against the CPU results under Linux Fedora Core 6. We have only evaluated the computational time without the time needed to read data from disk.

We focus on numerical tests with large datasets from on the UCI repository, including Forest cover type, KDD cup 1999 and Adult datasets (c.f. table 1). We created another massive datasets by using the RingNorm program. It is a 20 dimensional, 2 class classification example. Each class is drawn from a

Table 1. Dataset description

Datasets	Dimensions	Training set	Testing set
Adult	110	32561	16281
Forest covertype	54	495141	45141
KDD cup 1999	41	4898429	311029
Ringnorm 1M	20	1000000	100000
Ringnorm 10M	20	10000000	1000000

Table 2. Classification results reported on a CPU (Intel Core 2, 2.6 GHz, 2 GB RAM) and a GPU (NVIDIA Geforce 8800 GTX)

Datasets	GPU time (sec)	CPU time (sec)	Accuracy (%)
Adult	0.48	17.52	85.18
Forest covertype	2.42	84.17	77.18
KDD cup 1999	18.01	552.98	92.31
Ringnorm 1M	0.39	39.01	75.07
Ringnorm 10M	17.44	395.67	76.68

multivariate normal distribution. Class 1 has mean equal to zero and covariance 4 times the identity. Class 2 (considered as -1) has unit covariance with mean = 2/sqrt(20).

First, we have split the datasets into small blocks of rows to avoid fitting in memory. Table 2 presents the classification results obtained by GPU and CPU implementations of the incremental Newton SVM algorithm. The GPU version is a factor of 45 faster than the CPU implementation.

For Forest cover type dataset, the standard LibSVM ran for 21 days without any result. Recently-published results indicate that the SVM-perf algorithm performed this classification in 171 seconds (CPU time) on a 3.6 GHz Intel Xeon processor with 2 GB RAM. This indicates that our GPU implementation of incremental Newton SVM is probably about 70 times faster than SVM-Perf.

KDD Cup 1999 dataset consists of network data indicating either normal connections (negative class) or attacks (postive class). LibSVM ran out of memory. CB-SVM has classified the dataset with over 90% accuracy in 4750 seconds (CPU time) on a Pentium 800 MHz with 1GB RAM, while our algorithm achieved over 92% accuracy in only 18.01 second. They appear to be about a factor of 264 times faster than CB-SVM.

The numerical test results showed the effectiveness of the new algorithm to deal with very large datasets on GPUs.

6 Conclusion and Future Work

We have presented a new parallel incremental Newton SVM algorithm being able to deal with very large datasets in classification tasks on GPUs. We have

extended the recent Newton SVM algorithm proposed by Mangasarian in two ways. We developed an incremental algorithm for classifying massive datasets. Our algorithm avoid loading the whole dataset in main memory: only subsets of the data are considered at any one time and update the solution in growing training set. We developed a parallel version of incremental Newton SVM algorithm based on GPUs to gain high performance at low cost.

We evaluated the performances in terms of learning time on very large datasets of UCI repository and Delve. The results showed that our algorithm using GPU is about 45 times faster than a CPU implementation. We also compared the performances of our algorithm with the efficient standard SVM algorithm LibSVM and with two recent algorithms, SVM-perf and CB-SVM. Our GPU implementation of incremental Newton SVM is probably over 100 times faster than LibSVM, SVM-Perf and CB-SVM.

A forthcoming improvement will extend our methods for dealing with complex non-linear classification tasks.

References

1. Blake, C., Merz, C.: UCI Repository of Machine Learning Databases (2008)
2. Boser, B., Guyon, I., Vapnik, V.: An Training Algorithm for Optimal Margin Classifiers. In: Proc. of 5th ACM Annual Workshop on Computational Learning Theory, Pittsburgh, Pennsylvania, pp. 144–152 (1992)
3. Cauwenberghs, G., Poggio, T.: Incremental and Decremental Support Vector Machine Learning. In: Advances in Neural Information Processing Systems, vol. 13, pp. 409–415. MIT Press, Cambridge (2001)
4. Chang, C.C., Lin, C.J.: LIBSVM – A Library for Support Vector Machines (2001)
5. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
6. Delve: Data for evaluating learning in valid experiments (1996)
7. Do, T.N., Poulet, F.: Towards High Dimensional Data Mining with Boosting of PSVM and Visualization Tools. In: Proc. of 6th Int. Conf. on Entreprise Information Systems, pp. 36–41 (2004)
8. Do, T.N., Poulet, F.: Mining Very Large Datasets with SVM and Visualization. In: Proc. of 7th Int. Conf. on Entreprise Information Systems, pp. 127–134 (2005)
9. Do, T.N., Poulet, F.: Classifying one billion data with a new distributed SVM algorithm. In: Proc. of 4th IEEE International Conference on Computer Science, Research, Innovation and Vision for the Future, pp. 59–66 (2006)
10. Do, T.N., Fekete, J.D.: Large Scale Classification with Support Vector Machine Algorithms. In: Proc. of 6th International Conference on Machine Learning and Applications, pp. 7–12. IEEE Press, USA (2007)
11. Dongarra, J., Pozo, R., Walker, D.: LAPACK++: a design overview of object-oriented extensions for high performance linear algebra. In: Proc. of Supercomputing 1993, pp. 162–171. IEEE Press, Los Alamitos (1993)
12. Fayyad, U., Piatetsky-Shapiro, G., Uthurusamy, R.: Summary from the KDD-03 Panel - Data Mining: The Next 10 Years. SIGKDD Explorations 5(2), 191–196 (2004)

13. Fung, G., Mangasarian, O.: Incremental Support Vector Machine Classification. In: Proc. of the 2nd SIAM Int. Conf. on Data Mining SDM, USA (2002)
14. Guyon, I.: Web Page on SVM Applications (1999)
15. Joachims, T.: Training Linear SVMs in Linear Time. In: Proc. of the ACM SIGKDD Intl Conf. on KDD, pp. 217–226 (2006)
16. Lyman, P., Varian, H.R., Swearingen, K., Charles, P., Good, N., Jordan, L., Pal, J.: How much information (2003)
17. Mangasarian, O.: A finite newton method for classification problems. Data Mining Institute Technical Report 01-11, Computer Sciences Department, University of Wisconsin (2001)
18. Mangasarian, O., Musicant, D.: Lagrangian Support Vector Machines. Journal of Machine Learning Research 1, 161–177 (2001)
19. NVIDIA CUDA: CUDA Programming Guide 1.1 (2007)
20. NVIDIA CUDA: CUDA CUBLAS Library 1.1 (2007)
21. Osuna, E., Freund, R., Girosi, F.: An Improved Training Algorithm for Support Vector Machines. Neural Networks for Signal Processing VII, 276–285 (1997)
22. Platt, J.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning, pp. 185–208 (1999)
23. Poulet, F., Do, T.N.: Mining Very Large Datasets with Support Vector Machine Algorithms. In: Camp, O., Filipe, J., Hammoudi, S., Piattini, M., et al. (eds.) Enterprise Information Systems V, pp. 177–184. Kluwer Academic Publishers, Dordrecht (2004)
24. Syed, N., Liu, H., Sung, K.: Incremental Learning with Support Vector Machines. In: Proc. of the 6th ACM SIGKDD Intl Conf. on KDD 1999, USA (1999)
25. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. In: Proc. of 17th Int. Conf. on Machine Learning, pp. 999–1006 (2000)
26. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
27. Wasson, S.: Nvidia's GeForce 8800 graphics processor. Technical report, PC Hardware Explored (2006)
28. Yu, H., Yang, J., Han, J.: Classifying large data sets using SVMs with hierarchical clusters. In: Proc. of the ACM SIGKDD Intl Conf. on KDD, pp. 306–315 (2003)

A Wavelet Based Multi Scale VaR Model for Agricultural Market

Kaijian He[1,2], Kin Keung Lai[1], Sy-Ming Guu[3], and Jinlong Zhang[4]

[1] Department of Management Sciences, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong
paulhekj@cityu.edu.hk, mskklai@cityu.edu.hk
[2] College of Business Administration, Hunan University, Changsha, Hunan, 410082, P.R. China
[3] College of Management, Yuan-Ze University, 135 Yuan-Tung Road, Chung-Li, Taoyuan, 32003, Taiwan
iesmguu@saturn.yzu.edu.tw
[4] College of Management, Huazhong University of Science and Technology, Wuhan, 430074, P.R. China
jlzhang@mail.hust.sc.cn

Abstract. Participants in the agricultural industries are subject to significant market risks due to long production lags. Traditional methodology analyzes the risk evolution following a time invariant approach. However, this paper analyzes and proposes wavelet analysis to track risk evolution in a time variant fashion. A wavelet-econometric hybrid model is further proposed for VaR estimates. The proposed wavelet decomposed VaR (WDVaR) is ex-ante in nature and is capable of estimating risks that are multi-scale structured. Empirical studies in major agricultural markets are conducted for both the hybrid ARMA-GARCH VaR and the proposed WDVaR. Experiment results confirm significant performance improvement. Besides, incorporation of time variant risks tracking capability offers additional flexibility for adaptability of the proposed hybrid algorithm to different market environments. WDVaR can be tailored to specific market characteristics to capture unique investment styles, time horizons, etc.

Keywords: financial, risk management, time series analysis, wavelets and fractals, Value at Risk.

1 Introduction

Risks are an inherent part of agricultural production process due to the complexities in the surrounding physical and economic environment. Proper measurement and management of agricultural market risks are essential due to the following reasons: Firstly, the past and present risk levels shape expectations about future risk evolutions and influence production decisions. Secondly, risk levels affect important operational decisions concerning the cost of capital and revenue targets, etc. Thirdly, agricultural industries are capital intensive, with majority of

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 429–438, 2008.

capital deployment concentrated on farm real estate and machinery. Fourthly, agricultural risks increase continuously as there are rising levels of uncertainties during the production cycle. Thus, the industries are increasingly vulnerable to devastating consequences of unexpected market risks [1]. Value at Risk (VaR) is one popular approach to measure market risk. Despite its significance, there are only a handful of research methodologies concerning quantitative measurement and management of risks in agricultural industries - e.g. Giot measures risks in commodities markets, including metals, agricultural commodities and oil markets, using the VaR methodology. VaR estimated by using the APARCH model provides the highest reliability. VaR estimates based on implied volatility are also found to provide comparable performance [2,3]. Ani and Peter applied two popular credit risk models to risk measurement in agricultural loans and calculated the required VaR to protect investors' interests [4]. At the same time, measurement of the multi-scale heterogeneous structure of agricultural risk evolution remains the unexplored area.

Therefore, this paper proposes an ex-ante decomposition based approach for risk measurement in agricultural markets, in contrast with the traditional approaches. Wavelet analysis is proposed to conduct multi-resolution analysis of the heterogeneous market structure of the risk evolution process in agricultural markets. Wavelet analysis has been used extensively in different fields of economics and finance, such as the economic relationship identification and wavelet decomposed forecasting methodology, etc [5,6,7]. Despite the apparent need for multi-scale risk structure analysis, there have been only a handful of researches identified in the literature. These approaches focus on multi-resolution analysis of historical market risk structure and its distribution [8]. However, they are more of a historical simulation approach during their modeling attempts and offer little insights into the evolution of these structures.

The proposed Wavelet based approach for VaR estimates allows the flexibility of combining the power of different econometric models in the time scale domain, which reflects different investment strategies over various investment time horizons in the agricultural markets.

Empirical studies have been conducted in major US agricultural markets to evaluate and compare the performance of the proposed wavelet based approach against the traditional ARMA-GARCH approach for VaR estimates. Experiment results confirm improved reliability and accuracy offered by WDVaR due to its ability to analyze multi-scale heterogeneous structures and its processing power.

The rest of the paper is organized as follows: the second section briefly reviews the relevant theories, including wavelet analysis and different approaches to VaR estimates. The status quo of applications of wavelet analysis in risk management is also reviewed. The third section proposes the wavelet based VaR algorithm. The fourth section conducts empirical studies in major US agricultural markets. Performance evaluation and comparison of models tested are based on Kupiec backtesting procedures. The fifth section concludes.

2 Relevant Theories

2.1 Value at Risk

Value at Risk is the dominant risk measure that has received endorsement from both, academics and industries, recently [9]. Given the confidence level α, VaR is defined as the p-quantile of the portfolio's profit/loss distribution over certain holding period at time t as in (1).

$$VaR_t = -q_{p,t} = F_t^{-1}(\alpha) = \mu_t + \sigma_t G_t^{-1}(\alpha) \tag{1}$$

Where $q_{p,t}$ refers to the p^{th} conditional quantile of the portfolio distribution. $F_t^{-1}(\alpha)$ and $G_t^{-1}(\alpha)$ refer to the inverse of the portfolio distribution function. μ_t is the conditional mean, while σ_t is the conditional variance. VaR is used to compress and give approximate estimates as to the maximal possible losses.

Estimation of VaR can be classified into three groups, depending on the degree of assumptions made - parametric approach, non-parametric approach and semi-parametric approach [9]. The parametric approach fits the curve into the risk evolution process and derives analytical forms. The advantage of the parametric approach is its intuitive appeal and simplicity to understand and track. It is especially useful in the tranquile environments. However, when the market gets volatile with more extreme events occurring, assumptions in parametric approaches are easily violated and lead to biased estimates. Also, the current parametric approaches lack the essential ability to analyze the multi-scale nonlinear dynamics in the markets. The non-parametric approach takes a different route by imposing weak assumptions during the estimation process. It includes techniques such as Monte Carlo simulation methods and more recently neural network, etc. The advantage lies in its adaptability to non-linear environments, where Data Generating Process (DGP) is unknown. However, These approaches are mostly black box in nature and offer little insights into the underlying risk evolutions. The semi-parametric approach strikes the balance between the previous two approaches. It relaxes to some extent, assumptions made in parametric approaches while providing more insights. Techniques used include extreme value theory (EVT) and wavelet analysis, etc.

Backtesting procedures are formal statistical methods to verify whether the projected losses are in line with the actual losses observed in the market. Over the years, different approaches have been developed. This paper uses the unconditional coverage tests as the basis for the model evaluation and comparison. The VaR exceedances are Bernoulli random variable, which is equal to 1 when VaR is exceeded by losses; and is 0 otherwise. The null hypothesis for unconditional coverage test is the acceptance of the model at the given confidence level. Kupiec develops the likelihood ratio test statistics as in (2).

$$LR = -2ln[(1-\rho)^{n-x}\rho^x] + 2ln[(1-x)/n)^{n-x}(x/n)^x] \tag{2}$$

Where x is the number of exceedances, n is the total number of observations and p is the confidence level chosen for the VaR estimates. The Kupiec likelihood ratio test statistics is distributed as $\chi^2(1)$.

2.2 Wavelet Analysis

Wavelet analysis with time-frequency localization capability is introduced as the advancement to the traditional band limited Fourier transform [10]. The wavelet functions utilized during wavelet analysis are mathematically defined as functions that satisfy the admissibility condition as in (3).

$$C_\psi = \int_0^\infty \frac{|\varphi(f)|}{f} df < \infty \tag{3}$$

Where φ is the Fourier transform of the frequency f. ψ is the wavelet transform.

Location parameter u and scale parameter s can be used to translate and dilate the original function during wavelet analysis as in (4) [10].

$$W(u,s) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{s}} \psi(\frac{t-u}{s}) dt \tag{4}$$

Where $s \in R^+, \mu \in R$. Thus, the transformed wavelet function convolves with the market return series to obtain wavelet coefficients as in (4)

An inverse operation could also be performed, as in (5), which is referred to as the wavelet synthesizing process.

$$x(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^{\infty} W(u,s) \psi_{u,s}(t) du \frac{ds}{s^2} \tag{5}$$

By design, wavelets are dilated shorter at higher frequency, which provides shorter time windows to capture time sensitive information. Wavelets are also dilated longer at lower frequency to emphasize frequency level information with longer time windows. Typical wavelet families include Haar, Daubechies, Symlet and Coiflet, etc.

3 Wavelet Decomposed Value at Risk

Markets are heterogeneous in nature. A typical financial market (e.g. agricultural, energy market, etc.) consists of the following participants: market makers, intraday traders, daily traders, short term traders and long term traders. Since different types of traders have different investment strategies, determined by their investment horizon and financial health, their contributions to the market price formation process vary in terms of both time horizon and frequency level [11]. Previous parametric approaches, including single and hybrid models based approaches to VaR estimates, are categorized as ex-post approaches. They would lead to significant biases in estimates in heterogeneous markets. In heterogeneous markets, prices are formed with influences from different types of investors, characterized by different investment strategies and time horizons, which change over time. Since most current single parametric approaches are based on stationary assumptions and focus on frequency domain, their performances are unstable in the volatile market environment, i.e. the violation of the stationary assumptions

invalidates the model during periods of intense fluctuations with investment strategies changing with time horizons. The ensemble approaches also share the same problem with the single model approach, although it improves the performance by nonlinearly ensembling different individual forecasts. Besides, it offers little insights into the multi-scale market risk structure. Meanwhile, the current hybrid algorithm linearly filters the data through different models to extract maximal possible information and minimize the residuals. However, the bias introduced in the first stage filtering process would not only be carried forward, but would also distort the next filtering process and lead to an increasing level of biases in estimates.

If the distribution can be described with location and scale parameters, then the VaR is estimated parametrically as in (6)

$$VaR_t = \mu_t + \sigma_t G^{-1}(\alpha) \tag{6}$$

Where $G^{-1}(\alpha)$ refers to the inverse of the cumulative normal distribution.

Estimation of the conditional mean μ_t follows the ARMA-GARCH process. Estimation of the conditional standard deviation σ_t follows a multi-scale framework based on wavelet analysis.

Firstly , the original data series are projected into the time scale domain with the chosen wavelet families as in (7)

$$f(t) = f_{A^J}(t) + \sum_{j=1}^{J} f_{D^j}(t) \tag{7}$$

Where $f(t)$ refers to the original time series. $f_{A^J}(t)$ refers to the decomposed time series using scaling function at scale J. $f_{D^j}(t)$ refers to the decomposed time series using wavelet function at scales j, up to scale J.

Secondly econometric or time series models serve as individual volatility forecasters at each scale. Parameters are estimated by fitting different models to decomposed data at each scale. Then volatilities are forecasted, using the estimated model specifications.

Thirdly, according to the preservation of energy property, estimates of volatility are reconstructed from volatility estimates at each scale, using wavelet synthesis techniques as in (8).

$$\begin{aligned} \widehat{\sigma}^2 = VaR((f(t)) &= var(f_{A^J}(t)) + \sum_{j=1}^{J} Var(f_{D^J}(t)) \\ &= \frac{1}{2\lambda_J \widehat{N}} \sum_{t=2}^{\frac{N}{2^J-1}} \omega_{J,t}^2 + \sum_{j=1}^{J} \frac{1}{2\lambda_j \widehat{N}} \sum_{t=2}^{\frac{N}{2^j-1}} \varphi_{j,t}^2 \end{aligned} \tag{8}$$

Where $N = 2^J$ refers to the length of the dyadic data series. $\widehat{N}$ refers to the wavelet coeeficients at scale λ_j.

4 Empirical Studies

4.1 Data and Experiment Design

The data examined in this paper are daily aggregated spot prices in two major US agricultural markets: cotton and live hog. These markets are selected, based on their significant market shares and data availability. The market shares for both cotton and live hog are 2.20% and 5.90% respectively. The data set for cotton covers the time period from 27 March, 1980 to 14 June, 2006 while the data set for live hog covers the time period from 2 January, 1980 to 14 June, 2006. The total sample size is 6613 daily observations. The data are divided into two parts, i.e., the first 60% of the data set forms the training set while the rest 40% serves as the test set.

Table 1. Descriptive Statistics and Statistical Tests

Agricultural Commodities	Cotton	Live Hog
Mean	0.0000	0.0000
Maximum	0.1292	0.5125
Minimum	-0.9049	-0.5316
Medium	0.0000	0.0000
Standard Deviation	0.0196	0.0248
Skewness	-24.7588	-0.3208
Kurtosis	1150.6933	108.1585
Jarque-Bera Test (P value)	0	0
BDS Test (P value)	0	0

Table 1 reports the descriptive statistics for daily returns in both markets. The agricultural market represents a volatile environment, as indicated by the high volatility level. Investors face significant losses, as suggested by the negative and skewnesses. The market environment is considerably risky, as indicated by the high degree of excess kurtosis, which suggests the prevalence of extreme events. Thus, proper measurement and management of risks are crucial to both, investors and governments, in agricultural markets. Rejection of Jarque-bera test of normality and BDS (Brock-Dechert-Scheinkman) test of independence suggests the existence of nonlinear dynamics in the data [12]. Further performance improvement upon traditional approaches demands innovative techniques to account for the multi-scale heterogeneous structure of the markets.

A portfolio of one asset position worth 1 USD is assumed during each experiment. Geometric returns r_t are calculated assuming continuous compounding as $ln\frac{P_t}{P_{t-1}}$, where P_t refers to the price at time t. Based on the analysis of the autocorrelation and partial autocorrelation function, the model order for ARMA-GARCH is determined as ARMA(2,2) and GARCH(1,1). The length of the moving windows during the one step ahead forecasts is set at 3967 to cover the most relevant information set.

4.2 Empirical Results

ARMA-GARCH VaR. As suggested by results in table 2, the ARMA-GARCH approach don't offer sufficient reliability for VaR estimates. The ARMA-GARCH approach is rejected uniformly in both markets, across all confidence levels. Generally, the estimates are too conservative. The poor performance stems from the linear hybrid approach adopted, which lacks the ability to extract information concerning the multi-scale risk structures.

WDVaR(X,1). When wavelet analysis is applied to multi-resolution analysis of the risk evolution, two new parameters are introduced in the notion WDVaR(X,i), i.e., the wavelet families chosen X and the decomposition level i.

The sensitivity of the model's performance to the wavelet families chosen is investigated by estimating VaRs based on different wavelet families at the decomposition level 1. Experiment results are listed in table 3.

Taking VaRs estimated at 95% confidence level, experiment results in table 3 confirm that the wavelet families chosen affect the perspectives taken during the analysis of the risk evolution and, as a consequence, affect the VaR estimated. The wavelet families could be treated as a pattern recognition tool since different families would lead to the extraction of different data patterns. Convolution of wavelets to the original data series is a process of searching for the relevant data patterns across time horizons and scales. Meanwhile, further experiment results confirm that

Take symlet 2 for example, experiment results in table 4 show that changing wavelet families do lead to significant performance improvement. VaRs estimated are accepted at both 97.5% and 99% confidence level in the live hog market and are accepted at both 95% and 97.5% confidence levels in the cotton market.

WDVaR(Haar, i). The sensitivity of the model's performance to the selection of decomposition level is further investigated. Decomposition level is set to 3.

Increases in the decomposition level improve the model's performance significantly. Firstly, the reliability of the proposed VaRs estimates are accepted at

Table 2. Experiment Results for ARMA-GARCH VaR in Two Agricultural Commodities Markets Across All Confidence Levels

Agricultural Commodities	Confidence Level	ARMA-GARCH VaR Exceedance	MSE	Kupiec Test Statistics	P-value
Cotton	99.0%	8	0.0018	17.8826	0
	97.5%	17	0.0013	52.9588	0
	95.0%	31	0.0010	116.5033	0
Live Hog	99.0%	15	0.0027	5.9125	0.0149
	97.5%	33	0.0020	20.7756	0
	95.0%	71	0.0015	35.6069	0

Table 3. Experiment Results for WDVaR(X,1) at 95% Confidence Level in Cotton Market

Wavelet Family	WDVaR(x,1) Exceedance	MSE	Kupiec Test Statistics	P-Value
Haar(db1)	45	0.0007	80.4040	0.0000
db2	113	0.0005	3.0806	0.0792
db3	141	0.0005	0.6041	0.4370
db4	131	0.0005	3.0806	0.0792
db5	143	0.0004	0.9057	0.3413
db6	151	0.0004	2.6962	0.1006
dmey	138	0.0004	0.2642	0.6072
sym2	113	0.0005	3.0806	0.0792
sym3	141	0.0005	0.6041	0.4370
sym4	135	0.0005	0.0620	0.8033
sym5	132	0.0004	0.0003	0.9858
coif1	148	0.0005	1.9169	0.1662

Table 4. Experiment Results for WDVaR(Sym2,1) in Two Agricultural Commodities Markets Across All Confidence Levels

Agricultural Commodities	Confidence Level	ARMA-GARCH VaR Exceedance	MSE	Kupiec Test Statistics	P-value
Cotton	99.0%	57	0.0008	26.8099	0
	97.5%	79	0.0006	2.4328	0.1188
	95.0%	113	0.0005	3.0806	0.0792
Live Hog	99.0%	25	0.0027	0.0807	0.7764
	97.5%	51	0.0020	3.8353	0.0502
	95.0%	97	0.0016	10.8276	0

99% confidence level in the cotton market and are accepted at both 97.5% and 99% confidence levels in the live hog market. Secondly, the accuracy of the estimates improves as the size of the exceedances measured by Mean Square Error (MSE) decreases uniformly. This performance improvement results from finer modeling of details at higher decomposition levels using the wavelet analysis. As the decomposition level increases, market structures are projected into the higher dimension domain to reveal more subtle details, i.e. investors with longer investment horizons are separated out for further analysis. Thus, the attempted ARMA-ARCH model could be estimated with more suitable parameters at the individual scale, which results in the more accurate description of risk evolutions. Aggregated together, it will result in the closer tracking of risk evolutions and,

Table 5. Experiment Result for WDVaR(Haar,3) in Two Agricultural Commodities Markets Across All Confidence Levels

Agricultural Commodities	Confidence Level	ARMA-GARCH VaR Exceedance	MSE	Kupiec Test Statistics	P-value
Cotton	99.0%	21	0.0011	1.2164	0.2701
	97.5%	36	0.0008	16.7993	0
	95.0%	58	0.0006	55.0014	0
Live Hog	99.0%	29	0.0026	0.2427	0.6222
	97.5%	55	0.0020	2.0258	0.1546
	95.0%	94	0.0015	12.8661	0

thus, more accurate and reliable estimates of risk measurements - VaR. Besides, by tuning the two new parameters, i.e. the wavelet families and the decomposition level, reliability and accuracy of the VaR estimates improves significantly. The wavelet based approach offers considerably more flexibility during the VaR estimation process. However, the increased performance doesn't come without costs. More subtleties are revealed with the exponential growth of computational complexities, which are not always desirable.

5 Conclusion

Given the long production cycle and unexpected factors involved, proper measurement of risks has a significant impact on agricultural production decisions and the revenue generated. This paper proposes the wavelet based hybrid approach to measure agricultural risks using the VaR methodology, due to its long production cycle. The contribution of this paper is two fold. Firstly, multiresolution analysis is conducted to investigate the heterogeneous market structures using the wavelet analysis. Agricultural data are projected into the time scale domain to reveal its the composition factors. Secondly, the ex-ante based methodology is proposed for hybrid algorithm design. Wavelet analysis is used as an example of ex-ante based hybrid algorithm. The combination methodology is based on time scale decomposition in contrast with the traditional linear filtering process. Experiments conducted in two major US agricultural markets show that the proposed WDVaR outperforms the traditional ARMA-GARCH VaR. The advantage of this model is that the estimates unify different models with different parameter settings in a given time scale domain. Besides, this model also offers additional insights into the multi-scale structure of risk evolution.

Acknowledgement

The work described in this paper was supported by a grant from the National Social Science Foundation of China (SSFC No.07AJL005) and a Research Grant of City University of Hong Kong (No. 9610058).

References

1. Aviles-Cano, M., Gonzalez-Estrada, A., Martinez-Damian, M.A.: Risk analysis, optimal portfolios and diversification in agriculture. Agrociencia 40, 409–417 (2006)
2. Giot, P., Laurent, S.: Market risk in commodity markets: a var approach. Energy Econ. 25, 435–457 (2003)
3. Giot, P.: The information content of implied volatility in agricultural commodity markets. J. Futures Mark. 23, 441–454 (2003)
4. Katchova, A.L., Barry, P.J.: Credit risk models and agricultural lending. Am. J. Agr. Econ. 87, 194–205 (2005)
5. Fernandez, V.: Does domestic cooperation lead to business-cycle convergence and financial linkages? Q. Rev. Econ. Financ. 46, 369–396 (2006)
6. Struzik, Z.R.: Wavelet methods in (financial) time-series processing. Physica A 296, 307–319 (2001)
7. Yousefi, S., Weinreich, I., Reinarz, D.: Wavelet-based prediction of oil prices. Chaos Solitons Fractals 25, 265–275 (2005)
8. Fernandez, V.: The CAPM and value at risk at different time-scales. Int. Rev. Financ. Anal. 15, 203–219 (2006)
9. Dowd, K.: Measuring market risk, 2nd edn. John Wiley & Son, West Sussex (2002)
10. Percival, D.B., Walden, A.T.: Wavelet methods for time series analysis. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge (2000)
11. Gencay, R., Selcuk, F., Whitcher, B.: An introduction to wavelets and other filtering methods in finance and economics. Academic Press, San Diego (2002)
12. Brock, W.A., Hsieh, D.A., LeBaron, B.D.: Nonlinear dynamics, chaos, and instability: statistical theory and economic evidence. MIT Press, Cambridge (1991)

Using Formal Concept Analysis for the Extraction of Groups of Co-expressed Genes

Mehdi Kaytoue-Uberall[1], Sébastien Duplessis[2], and Amedeo Napoli[1]

[1] LORIA – Campus Scientifique B.P. 239 – 54506
Vandoeuvre-lès-Nancy Cedex, France
mehdi.kaytoueuberall@loria.fr
[2] INRA UMR 1136 – Interactions Arbres/Microorganismes –
54280 Champenoux, France

Abstract. In this paper, we present a data-mining approach in gene expression matrices. The method is aimed at extracting formal concepts, representing sets of genes that present similar quantitative variations of expression in certain biological situations or environments. Formal Concept Analysis is used both for its abilities in data-mining and information representation. We structure the method around three steps: numerical data is turned into binary data, then formal concepts are extracted and filtered with a new formalism. The method has been applied to a gene expression dataset obtained in a fungal species named *Laccaria bicolor*. The paper ends with a discussion and research perspectives.

Keywords: Gene expression, formal concept analysis, scaling.

1 Introduction

Microarray biotechnologies can monitor the expression of thousands of genes across many biological situations, over time, and have proved being relevant in many applications. They allow to classify tumors or tissue types, to identify genes that play a major role in a given cellular process, or eventually to assign a function to a gene [18].

The output of a microarray experiment is a matrix or table with genes in lines and biological situations in columns (see Table 1). Each value of this so-called gene expression matrix (GEM) reflects the state of expression (transcription) of a gene in a given situation. A classical method for analysing GEM is clustering, which groups into clusters, genes that exhibit similar expression patterns in all the different situations. Indeed a consensual hypothesis in molecular biology states that co-expressed genes, i.e. genes having a similar expression pattern, interact together within the same biological function or the same cellular process [10,19]. For analysing GEM, biologists apply widely used classical numerical methods such as K-Means, hierarchical clustering, and Self Organizing Maps [8]. Meanwhile, symbolic data-mining methods [14] such as itemset search [15], association rules extraction [5], and Formal Concept Analysis [17], are emerging thanks to their ease of result interpretation.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 439–449, 2008.

In this paper, we propose a method relying on Formal Concept Analysis (FCA) [6] for extracting groups or classes of co-expressed genes. FCA builds a concept lattice where each concept represents a set of co-expressed genes in a number of situations. As input data for FCA is a binary table, a numerical GEM is transformed with possible introduction of biases or loss of information. Moreover, the set of resulting concepts can be huge (up to a million) and contains a little proportion of biologically relevant concepts for a given study [17]. Indeed, most of the concepts characterize groups of genes showing a similar expression pattern, with very low numerical variation between the situations. The reason is that a few proportion of genes has its expression differing from one situation to another, and that microarray data are noisy. In this paper, we propose an original transformation from numerical to binary GEM data. This transformation allows to characterize and easily discriminate groups of co-expressed genes that shows particular expression variations. The biologists are able to infer from this characterization the role of genes and their membership to a cellular process.

The paper is organized as follows. In Section 2, we explain what gene expression data are and why we need new methods for GEM analysis. After explaining the background on FCA, our approach is detailed in Section 3 and applied to a real dataset in Section 4. The paper closes with a discussion and research perspectives.

2 Background

2.1 Gene Expression Matrices and Profiles

Biological processes of a living cell are based on chemical reactions and interactions between molecules. Proteins are molecules playing a major role in structure and function of cells, tissues and the whole organism. They are produced by a blueprint encoded in the DNA. A portion of DNA called the coding sequence of a gene serves as support to build a specific protein. The mechanism that produces a protein from a gene is called gene expression. It consists of two steps: transcription and translation. During the transcription, a copy of a gene is produced, called messenger RNA (mRNA). Then the translation produces a protein from the mRNA. The different reactions and interactions of a cell differ from the quantity of each protein at a time. It is nowadays still hard and expensive to measure the abundance of several proteins in a cell. However, microarray is a less expensive biotechnology and enables the measurement of the abundance of thousands of different mRNA. The abundance of mRNA of a gene is measured into a numerical value called gene expression value. In this paper, we consider the NimbleGen Systems Oligonucleotide Arrays technology: expression values are ranged from 0 (not expressed) to 65535 (highly expressed).

A microarray experiment considers a large number of genes, eventually the complete coding space of a genome in multiple situations. These situations can be a time-series during a particular biological process (e.g. cell cycle), a collection of different tissues (e.g. normal and cancerous tissues) or both, sometimes responding to particular environmental stresses.

By measuring the expression value of a gene in m situations, a gene expression profile can be written as a m-dimensional numerical vector $e = (e_1, ..., e_m)$ where e_j is the expression value of the gene in the j^{th} situation ($j \in [1, m]$). A gene expression matrix (GEM) $E = (e_{ij})_{1 \leq i \leq n, 1 \leq j \leq m}$ is a collection of n profiles: it is composed of n lines which correspond to genes and m columns which corresponds to situations. e_{ij} is the expression value of the i^{th} gene in the j^{th} situation. For example, in Table 1, $(11050, 11950, 1503)$ is the expression profile for the Gene 1. $e_{11} = 11050$ is the expression value of the Gene 1 in the situation a. Clustering methods groups similar profiles together into a cluster, leading, when interpreted by a domain expert, to the understanding of biological processes and of function of genes [10,19].

Table 1. An example of GEM composed of 5 genes in lines and 3 situations in columns

Gene Id	a	b	c
Gene 1	11050	11950	1503
Gene 2	13025	14100	1708
Gene 3	6257	5057	6500
Gene 4	5392	6020	7300
Gene 5	13070	12021	15548

Why do we still need new methods? Although literature includes many methods for the analysis of GEM [8,11], the challenge to derive useful knowledge from GEM still remains. Indeed, biological background implies to take simultaneously the following properties into account:

1. A single gene can participate in several biological processes or have several functions.
2. A single situation can describe several biological processes or functions.
3. A biological function or process implies a small subset of genes.
4. A biological process or function of interest is active in several, all or none situations of a given dataset.
5. The genes having a high difference of expression value between two situations (e.g. between a normal and a cancerous cell) are not frequent.

A cluster represents a group of genes having globally similar expression values in all situations, e.g. Gene 1 and Gene 2 in Table 1. However, clustering methods may fail to detect strong local association between genes in some subsets of situations only, e.g. Gene 1, Gene 2 and Gene 5 are co-expressed in the situations a and b. Moreover most of clustering methods forces a profile to belong to only one cluster.

To overcome these limitations, the principle of block-clustering introduced in [7] has been adapted for GEM since [3]. These so-called bi-clustering methods are able to consider subsets of genes sharing compatible expression local patterns across subsets of biological situations (see Table 1). Some techniques allow genes

and conditions to belong to several bi-clusters. In this way, bi-clustering is better adapted for GEM analysis than clustering. Nevertheless a critical limit is that the potential number of interesting bi-clusters is huge (up to millions). A complete enumeration is not feasible in GEM analysis. Heuristics are introduced to reduce the result size but can miss bi-clusters of interest [11].

On the other hand, researchers considered algorithms in binary data [14] extracting local patterns, e.g. itemset extraction, association rules generation, and formal concept analysis [15,17,5,13]. As these methods works on a binary table, GEM has to be discretized, and then strong local patterns can be extracted. However, if the number of patterns is generally tractable in GEM analysis, it is too huge to by analysed by a human-expert. Some solutions exists to reduce the number of patterns and are presented in the related work section. We propose an alternative: to use Formal Concept Analysis (FCA) that is able to take into account properties 1, 2, 3, and 4. We present an original binary representation of the GEM that allows us to consider property 5, which, to our knowledge, is an unexplored area in FCA. This representation allows an expert to focus on and analyse most interesting patterns of the complete collection. Thus, we also exploit the abilities of FCA for data-mining and for information representation.

2.2 Background on FCA

Formal concept analysis (FCA) [6] is a mathematical formalism allowing to derive a concept lattice (to be defined later) from a formal context $\mathbb{K}$ constituted of a set of objects G, a set of attributes M, and a binary relation I defined on the Cartesian product $G \times M$. In the binary table representing $G \times M$ (see Table 2), the rows correspond to objects and the columns to attributes or properties. A cross means that "an object possesses a property". FCA can be used for a number of purposes among which knowledge formalization and acquisition, ontology design, and data mining.

The concept lattice is composed of *formal concepts*, or simply *concepts*, organized into a hierarchy by a partial ordering (a subsumption relation allowing to compare concepts). Intuitively, a concept is a pair (A, B) where $A \subseteq G$, $B \subseteq M$, and A is the maximal set of objects sharing the whole set of attributes in B and vice-versa. The concepts in a concept lattice are computed on the basis of a Galois connection defined by two derivation operators denoted by $'$:

$$' : 2^G \rightarrow 2^M; A' = \{m \in M; \forall g \in A : (g, m) \in I\}$$
$$' : 2^M \rightarrow 2^G; B' = \{g \in G; \forall m \in B : (g, m) \in I\}$$

Formally, a concept (A, B) verifies $A' = B$ and $B' = A$. The set A is called the *extent* and the set B the *intent* of the concept (A, B). The subsumption (or subconcept–superconcept) relation between concepts is defined as follows: $(A_1, B_1) \sqsubseteq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ (*or* $B_2 \subseteq B_1$). Relying on this subsumption relation $\sqsubseteq$, the set of all concepts, denoted by $\mathfrak{B}(G, M, I)$, extracted from a context $\mathbb{K} = (G, M, I)$ is organized within a complete lattice, that means that for any set of concepts there is a smallest superconcept and a largest subconcept, called the *concept lattice* of $\mathbb{K}$ and denoted by $\underline{\mathfrak{B}}(G, M, I)$.

The objects of microarray data are described by numerical attributes, i.e the expression value in each situation. FCA considers these attributes as *many-valued attributes*, in contrast to *one-valued* attributes. Then, a *many-valued context* (G, M, W, I) is a 4-tuple constituted of a set of objects G, a set of attributes M, a set of attribute values W and a ternary relation I defined on the Cartesian product $G \times M \times W$. $(g, m, w) \in I$, also written $g(m) = w$, means that "the value of the attribute m for the object g is w". The relation I verifies that $g(m) = w$ and $g(m) = v$ always imply $w = v$.

Before finding formal concepts in a many-valued context, this context has to be turned into a formal context (one-valued): many-valued attributes are discretized. This procedure is called discretization in data analysis, and termed also *conceptual scaling* in FCA.

3 Applying FCA to GEM Analysis

This section proposes to use FCA to extract from a GEM groups of co-expressed genes represented by concepts. Firstly, a GEM is mathematically defined as a many-valued context, then turned into a binary context using a particular conceptual scaling. The concepts of the formal context are searched for and structured into a concept lattice. Finally, concepts are filtered using a particular representation of concept intents.

A GEM as a many-valued context. A GEM is considered as a many valued context $\mathbb{K}_1 = (G, S, W, I_1)$ where G is a set of genes, S a set of situations, and $g(s) = w$ means that the expression value of gene g is w in situation s. In our example, $G = \{\text{Gene 1}, \ldots, \text{Gene 5}\}$, $S = \{a, b, c\}$, and I_1 is illustrated, for example, by Gene $1(a) = 11050$. The objectives are to use FCA to extract concepts (A, B), where $A \subseteq G$ is a subset of genes that shares similar values of W in the situations of $B \subseteq S$. As FCA needs a binary context, $\mathbb{K}_1$ is scaled.

Conceptual scaling. Given an attribute value space of the form $[0, u]$, the scale is given by a set of intervals $T = \{[0, u_1],]u_1, u_2], ...,]u_{p-1}, u_p]\}$. p is the number of intervals of T and $u_p = 65535$ for the NimbleGen System. In our context, the interval bounds for each $t \in T$ are dependent on expert knowledge. The scaling procedure consists in replacing each many-valued attribute of $\mathbb{K}_1 = (G, S, W, I_1)$ with p one-valued attributes to create the formal context $\mathbb{K}_2 = (G, S_T, I_2)$ with $S_T = S \times T$. S_T is then a set of pairs: the first value is a situation while the second represents an interval. $(g, (s, t)) \in I_2$ means that the gene g has an expression value in the interval t in the situation s.

This procedure is illustrated in the Table 2 with $T = \{[0, 5000[, [5000, 10000[, [10000, 65535]\}$. The many-valued attribute a is replaced by the three one-valued attributes (a, t_1), (a, t_2) and (a, t_3), i.e $(a, [0, 5000[)$, $(a, [5000, 10000[)$ and $(a, [10000, 65535])$. Then $(\text{Gene } 1, (a, t_3)) \in I_2$ means that Gene 1 has an expression value in $[10000, 65535]$ for the situation a and represented as the first cross in Table 2.

Classical discretization problems appear with conceptual scaling: introduction of biases and loss of information. Moreover, a major challenge in microarray analysis is to effectively dissociate actual gene expression values from experimental noise. We follow the idea given in [4,13] to characterize noise: a threshold $l \in [0, 1]$ is used to define the scale T as follows: $T = \{[0, u_1 + u_1 \times l], \ldots, [u_{p-1} - u_{p-1} \times l, u_p]\}$, meaning that intervals of T can overlap.

Table 2. Formal context derived from the many-valued context of Table 1

	(a,t_1)	(a,t_2)	(a,t_3)	(b,t_1)	(b,t_2)	(b,t_3)	(c,t_1)	(c,t_2)	(c,t_3)
Gene 1			×			×	×		
Gene 2			×			×	×		
Gene 3		×			×			×	
Gene 4		×			×			×	
Gene 5			×			×			×

Lattice construction. Once $\mathbb{K}_2 = (G, S_T, I_2)$ is computed, classical algorithms of lattice construction (see e.g. [2]) can be applied. The goal of such algorithms is to find the set of all concepts partially ordered by concept subsumption. In this paper, a concept (A, B) represents a subset of genes A that share similar expression values in the situations defined by the elements of B. The intent B is the common expression description of the genes in the extent A. For example, in Table 2, such a concept (A, B) is $(\{\textit{Gene 3, Gene 4}\}, \{(a, t_2), (b, t_2), (c, t_2)\})$. It means that Gene 3 and Gene 4 are co-expressed, by sharing expression values in the same interval t_2 in situations a, b and c.

The line diagram[1] of Figure 1 represents the concept lattice $\underline{\mathfrak{B}}(G, S_T, I_2)$. This is an ordered structure with a bottom and a top element. Top $= \{\{\textit{Gene 1, } \ldots, \textit{ Gene 5}\}, \{\emptyset\}\}$ and dually, Bottom has an empty extent and a maximal intent. Each element of the lattice represents a formal concept (A, B). The extent A is the set of objects attached to elements reachable by descending paths. Dually, the intent B is the set of attributes attached to elements reachable by ascending paths. For example, in Figure 1, the concept $C4 = \{\{\textit{Gene 1, Gene 2, Gene 5}\}, \{(a, t_3), (b, t_3)\}\}$.

This representation of the concept lattice is interesting for the biologists, because relations between the concepts provide knowledge: for example, concepts $C2, C3, C4$, and their relations show that Gene 1 and Gene 2 are co-expressed in all situations, but are also co-expressed with Gene 5 in situations a and b.

Concepts filtering. A GEM can contain thousands of genes and dozens of situations. For these reasons, the lattice $\underline{\mathfrak{B}}(G, S_T, I_2)$ may contain a large number of concepts (up to a million). The biologist focuses on small and homogeneous gene groups presenting the most important variations simultaneously. Interpretation of variations leads after experimental validations to the discovery of gene

[1] Drawn with the ConExp software, see http://conexp.sourceforge.net/

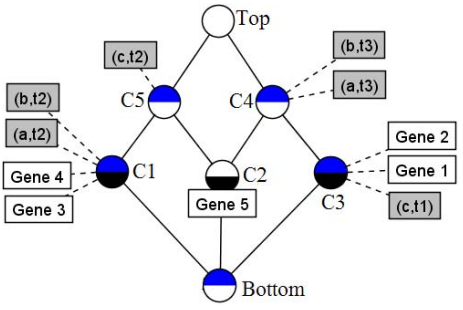


Fig. 1. The Concept lattice $\underline{\mathfrak{B}}(G, S_T, I_2)$

functions. Large variations are important to discriminate genes responsible of a particular cellular process [10]. Concepts are groups of genes co-expressed in a certain number of situations and satisfy the properties 1 and 2 proposed in Section 2: a gene (or a situation) may belong to multiple concepts. The properties 3 and 4 implies that a concept is a relevant bi-cluster if the extent is not composed of "too many" genes, and if the intent contains a least "a few" situations. A first filtering step keeps only concepts (A, B), with $|A| \leq a$ and $|B| \geq b$. a and b are chosen by the biologist and materialize the modalities "too many" and "a few". Property 5 implies that many concepts describe groups of co-expressed genes having a similar expression with no numerical variations, i.e. B is composed of pairs possessing the same interval $t \in T$. For example the concept $(\{Gene\ 3, Gene\ 4\}, \{(a, t_2), (b, t_2), (c, t_2)\})$ presents no variation, with respect to T. To take property 5 into account, instead of removing only the concepts that present no variation, we adapt the formalism proposed in [9]. For a concept, the intent B is a set of pairs (s, t). We replace t by its rank in T, the rank beginning at position 1. For example, $\{(a, t_2), (b, t_2), (c, t_2)\}$ becomes $\{(a, 2), (b, 2), (c, 2)\}$. B is now written as a set of pairs (s, k) where s remains a situation, while k is an integer valuation providing a control on expression values: $B = \{(a_1, k_1), \ldots, (a_p, k_p)\}$. We define $Var(B)$, which represents the number of variations of B. A variation is a non null difference between all the possible pairs of valuations of B (see below formula (1)). B is *constant*, i.e. presents no variation, iff $Var(B) = 0$. To have more control on variations, we also introduce $Var_\alpha(B)$ as the number of variations higher than a given threshold α, named minimal amplitude (see below formula (2)). It can be noticed that $Var_0(B) = Var(B)$. Finally, if B has β variations higher than α, then B is $(\alpha, \beta) - variant$ and respects $Var_\alpha(B) \geq \beta$. For example, the intent $B_3 = \{(a, 3), (b, 3), (c, 1)\})$ of the concept $C3 = (A_3, B_3)$, is $(2, 2) - variant$ as $Var_2(B_3) = |((a, 3), (c, 1)), ((b, 3), (c, 1))| = 2$.

$$Var(B) = |\{((a_i, k_i), (a_j, k_j))\ with\ i \neq j, |k_i - k_j| \neq 0, i, j \in [1, p]\}| \quad (1)$$

$$Var_\alpha(B) = |\{((a_i, k_i), (a_j, k_j))\ with\ i \neq j, |k_i - k_j| \geq \alpha, \alpha \geq 0, i, j \in [1, p]\}| \quad (2)$$

The biologists may introduce knowledge and preferences for a given biological study by filtering certain type of intents. A lot of combinations are possible, as k_i

is a value. We mainly extract concepts having $(\alpha, \beta) - variant$ intent to analyse groups of genes having the most important variations of expression profiles.

Related work. Conceptual scaling can be used to scale situations or genes. In this paper, we scale situations into u intervals. This allows us to extract relevant concepts by introducing filters. In [17], genes are scaled into a one-valued attribute depending on a threshold. Expression values greater than this threshold are said to be over-expressed. To overcome the problem of the number of concepts, the authors perform a clustering on concepts, where a distance between two concepts is considered. The authors of [1] use the same technique of discretization and extract association rules of type Gene 1 $\Rightarrow$ Gene 2, i.e. when the Gene 1 is over-expressed, so is the Gene 2. By using two thresholds, the authors of [5] extract association rules that follows the scheme: Gene 1 $\uparrow\Rightarrow$ Gene 2 $\downarrow$, i.e. when the expression of the Gene 1 is high, the expression of Gene 2 is low. We can notice that the above presented discretization methods are particular instances of the one presented in this paper. Recently FCA has been used for the comparison of situations in [4]. Firstly a concept lattice is built for each situation, and then a distance between two lattices is defined. It allows the authors to classify the situations. The authors of [13] use FCA to find biomarkers of a cancer (i.e. the set of genes involved) for supervised classification of cells.

4 Experiments

Biologists at INRA – UMR IAM[2], study the interactions between fungi and trees. They recently published the complete sequencing of the genome of a fungus called *Laccaria bicolor* [12]. This fungus can live in symbiosis with many trees of the temperate forest: the fungus grab mineral nutrients in surrounding soil, improve the nutrition of the tree, and receives carbon through association to the root tissue. This fungus is known to positively influence forest productivity. It is thus a major interest to understand how symbiosis performs at the cellular level.

The sequencing of *Laccaria bicolor* genome has allowed the prediction of more than 20,000 genes [12]. It remains now to study expression of those genes to understand functions and processes in the fungal lifestyle. Microarray measurements in several situations is a critical solution. For example, it enables to compare the expression values of genes in free-living cells with cells in a symbiotic association, to find genes responsible for the fructification of the fungus, etc.

A GEM is available as series (GSE9784) at the Gene Expression Omnibus at NCBI (http://www.ncbi.nlm.nih.gov/geo/). It is composed of 22,294 genes in lines and 7 various biological situations in columns, i.e. free-living cells (M81306 and MS238), young (FBe) and mature (FBl) fruiting body cells and fungal cells in association with trees (Mpgh, Mpiv and MD).

We have applied the method presented in this paper. The experiment starts with a many-valued context $\mathbb{K}_1 = (G, S, W, I)$ with $|G| = 22,294$ and $|S| = 7$. We have worked with a scale T where $t_1 = 100, t_2 = 250, t_3 = 500, t_4 = 1000,$

[2] Ecogenomic Team of the National Institute of Agronomical Research.

$t_5 = 2500, t_6 = 5000, t_7 = 7500, t_8 = 10000, t_9 = 12500, t_{10} = 15000, t_{11} = 17500, t_{12} = 20000, t_{13} = 30000, t_{14} = 40000$ and an overlapping threshold $l = 0.1$. The conceptual scaling produces the formal context $\mathbb{K}_2 = (G, S_T, I)$ where $|S_T| = |S| \times |T| = 98$. We remove all genes having all expression values in $[0, 100]$, i.e. a low intensity cut-off, considered by biologists as noise. Then $|G| = 17,862$. Extraction in $\mathbb{K}_2$ results in 146, 504 formal concepts. We filter out concepts (A, B) not respecting $|A| \leq 50$, $|B| \geq 4$ and B is a $(4, 4) - variant$ intent. It means that we extract groups of at most 50 genes that are co-expressed in at least 4 situations and showing at least 4 expression variations of a minimal amplitude greater than 4. We obtained 156 concepts which can be analysed by the expert. We pick up two concepts in Figure 2 and 3: the X-axis is composed of the elements of the intent B, the Y-axis is the expression value axis and a point (x, y) is the expression value of a gene in a situation. All points of a gene are linked: a line is the expression profile of a gene. In Figure 2, bold lines are average and standard deviation expression profiles.

The concept of Figure 2 is such that $|A| = 9$ and $|B| = 7$: 9 genes are strongly and globally (as $|B|$ is maximal) co-expressed. Considering the average expression profile of this group, biologist can infer that these genes are implied in the fungus fructification. Indeed expression value is higher in the early fruiting body that is the later one, very weak in free-living cells and higher in the cells where the association is well established (i.e. where fructification may appear).

In the concept of Figure 3, we have $|A| = 9$ and $|B| = 6$: 9 genes are strongly and locally (as $|B|$ is not maximal) co-expressed. These genes may have the same function or may belong to a single gene family but located at different chromosomal loci, with however an undefined biological function. The biologists show an active interest in this type of groups.

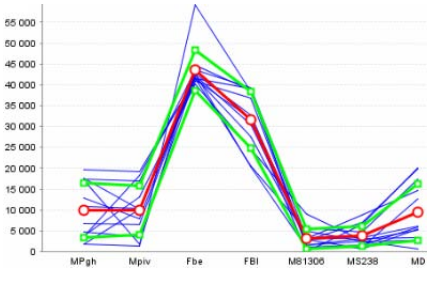


Fig. 2. A group of genes that may be involved in frutification of *Laccaria bicolor*

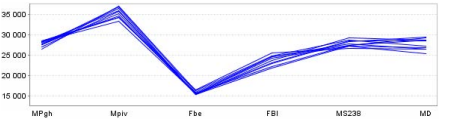


Fig. 3. A group of genes with coordinated expression profiles that can share similar function

5 Conclusion and Future Works

In this paper, we have shown how Formal Concept Analysis can be used to mine gene expression data and represent biological information. Indeed, an adapted and fully customizable conceptual scaling allows the expert to use knowledge to

filter the resulting formal concepts. However, from a qualitative point of view, there is no universal scaling. The impact of a scaling on the quality of the extracted formal concepts must be studied in each different case [16]. Moreover, a small percentage (less than 10% [8]) of genes manifest meaningful interest and are buried in large amount of noise. The present method allows us to extract strong local or global associations between genes and situations with the so-called formal concepts. The resulting set of concepts can be huge and the filtering may show its limits by skipping interesting concepts. The number of intervals of the scale may be critical for the number of final concepts. Solutions like formal concept clustering [17] will be studied, also to find a way to build a readable concept lattice for voluminous data. Finally, a binary context has to be enriched, when it is possible, with more reliable information on genes such as motifs on the up-sequence of DNA, putative function of genes... This will increase the reliability (not studied in this paper) by giving less importance on the gene expression values and thus on the inherent noise.

References

1. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.F., Gandrillon, O.: Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human sage data. Genome Biol. 3(12) (2002)
2. Berry, A., Bordat, J.-P., Sigayret, A.: A local approach to concept generation. Annals of Mathematics and Artificial Intelligence 49(1-4), 117–136 (2007)
3. Cheng, Y., Church, G.M.: Biclustering of expression data, pp. 93–103. ISMB (2000)
4. Choi, V., Huang, Y., Lam, V., Potter, D., Laubenbacher, R., Duca, K.: Using formal concept analysis for microarray data comparison. J Bioinform. Comput Biol. 6(1), 65–75 (2008)
5. Creighton, C., Hanash, S.: Mining gene expression databases for association rules. Bioinformatics 19(1), 79–86 (2003)
6. Ganter, B., Wille, R.: Formal Concept Analysis (1999)
7. Hartigan, J.A.: Direct clustering of a data matrix. J. Am. Statistical Assoc. 67(337), 123–129 (1972)
8. Jiang, D., Tang, C., Zhang, A.: Cluster analysis for gene expression data: a survey. IEEE Trans. on Knowledge and Data Engineering 16(11), 1370–1386 (2004)
9. Kaytoue-Uberall, M., Duplessis, S., Napoli, A.: Toward the discovery of itemsets with significant variations in gene expression matrices. SFC-CLADAG (2008)
10. Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.-B., Volkert, T.L., Fraenkel, E., Gifford, D.K., Young, R.A.: Transcriptional regulatory networks in saccharomyces cerevisiae. Science 298(5594), 799–804 (2002)
11. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
12. Martin, F., 67 other authors: The genome of laccaria bicolor provides insights into mycorrhizal symbiosis. Nature 452, 88–92 (2008)
13. Motameny, S., Versmold, B., Schmutzler, R.: Formal concept analysis for the identification of combinatorial biomarkers in breast cancer. In: ICFCA (2008)

14. Napoli, A.: A smooth introduction to symbolic methods for knowledge discovery. In: Handbook of Categorization in Cognitive Science, pp. 913–933 (2005)
15. Pensa, R.G., Besson, J., Boulicaut, J.-F.: A methodology for biologically relevant pattern discovery from gene expression data, pp. 230–241 (2004)
16. Rioult, F., Robardet, C., Blachon, S., Crémilleux, B., Gandrillon, O., Boulicaut, J.-F.: Mining concepts from large sage gene expression matrices. In: KDID, pp. 107–118 (2003)
17. Besson, J., Robardet, C., Boulicaut, J.-F., Gandrillon, O., Blachon, S., Pensa, R.G.: Clustering formal concepts to discover biologically relevant knowledge from gene expression data. Silico Biology 7(4–5), 467–483 (2007)
18. Stoughton, R.B.: Applications of dna microarrays in biology. Annual Review of Biochemistry 74(1), 53–82 (2005)
19. Stuart, J.M., Segal, E., Koller, D., Kim, S.K.: A gene-coexpression network for global discovery of conserved genetic modules. Science 302(5643), 249–255 (2003)

Join on Closure Systems Using Direct Implicational Basis Representation

Yoan Renaud

LIMOS - CNRS UMR 6158
Universit Blaise Pascal, Clermont-Ferrand
renaud@isima.fr

Abstract. Closure systems arise in many areas as databases, datamining, formal concept analysis, logic and artificial intelligence. Several representations were studied to deal efficiently with closure systems and to be efficient tools in various areas. Implicational basis is a particular representation which have the advantage to be a short representation of datas. This paper states on operation of join of closure systems using their implicational basis representations. Computation of an implicational basis of join of closure systems given by their implicational basis is a problem that can't be solve in polynomial time in size of the input in general. We present here a polynomial algorithm that solves this problem when the given implicational basis corresponding to the given closure systems are direct.

Keywords: closure systems, implicational basis, direct basis, datamining.

1 Introduction

Since several years, the amount of datas is growing in an exponential way and the storage becomes a serious difficulty. As a way, different representations of these datas has been studied to reduce their size. In particular, several representations were studied to deal efficiently with closure systems and to be efficient tools in various areas. Implicational basis is a particular representation which has the advantage to be a short representation of datas.

In this paper, we consider the issue of computing an implicational basis of join of closure systems given by their implicational basis. This may be encountered in many applications. Suppose you have to consider a large amount of datas and you want to analyse these datas and extract their corresponding implicational rules. A method would be to split these datas into a bounded number of packages and to compute in parallel their respective association rules. This step done, a process gets back the information generated and deduces the entire set of association rules. An other application concerns the updates of datas in a databasis. Suppose that we have the implicational rules concerning a set of datas and we add some datas, it will be more interresting to compute only the implicational rules corresponding to the new datas and use old ones than compute the implicational rules from all set of data. Then, computation of an implicational basis

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 450–457, 2008.

of join of closure systems given by their implicational basis would be interresting in that way if it could be obtained efficiently and if its size is reasonable.

Unfortunately, it has been shown in [7] that this problem can't be solve in polynomial time in the size of the input. As a way, we consider a specific class of basis: direct implicational basis as input and show that, in this case, the problem can be solved in polynomial time in size of the input.

The rest of this paper is organized as follows. In next section, we recall some basic definitions and introduce notations. In section 3, we define the problem of computing an implicational basis of join of closure systems given by their implicational basis. We show that, in general, there is no input polynomial algorithm. Then, we study the case where implicational basis are direct. Finally, in section 4, we conclude this paper.

2 Preliminaries

2.1 Basic Definitions and Notations

Let us review here only the most important concepts. A *partially ordered set* (poset) will be denoted by $P = (X, \leq_P)$, where X is the *ground set* of elements or vertices and $\leq_P$ is the order relation, i.e., an antisymmetric, reflexive and transitive binary relation whose elements $(a, b) \in \leq_P$ are written as $a \leq_P b$ $(a, b \in X)$ with the usual interpretation. If $a \leq_P b$ but $a \neq b$ then we write $a <_P b$. For $a, b \in X$ we say b *covers* a, denoted by $a \prec b$, if $a <_P b$ and there is no $c \in X$ with $a <_P c <_P b$. Two elements $a, b \in X$ are *comparable* in P (denoted by $a \sim_P b$) if $a \leq_P b$ or $b \leq_P a$. Otherwise they are said to be *incomparable* (denoted by $a \parallel_P b$).

With the notion of ordered sets we introduce the notion of lattice.

Definition 1. *Let L = (V, ≤) a non empty ordered set. L is a lattice if, for all x,y in V, x∨y and x∧y exist.*

Therefore a lattice contains a minimum element (according to the relation $\leq$) called the bottom of the lattice, and denoted $\perp$. Respectively, a lattice contains a maximum element called the top of the lattice, and denoted $\top$. We introduced specific elements of a lattice. An element j (respectively m) of a lattice L is a join-irreducible (respectively meet-irreducible) element of L if it canoot be obtained as the join (respectively meet) of elements of V all distinct from j (respectively from m). The sets of join-irreducible elements and meet-irreducible elements of a lattice L are respectively denoted by J_L and M_L. For an element x ∈ L, we denote by J_x (respectively M_x) the set of all join-irreducible elements j (respectively meet-irreducible elements m) such that j ≤ x (respectively x ≤ m).

2.2 Closure Systems

A set system on a set G is a family of subsets of G. A closure system $\mathcal{F}$ on a set G, also called a Moore family, is a set system closed under set intersection and which contains G.

Definition 2. *Let G be a finite set. A closure system on a set G is a family $\mathcal{F}$ of subsets of G, containing G and any intersection of subsets of $\mathcal{F}$, i.e. if it satisfies the following conditions :*

1. $G \in \mathcal{F}$.
2. $F, F' \in \mathcal{F}$ *implies* $F \cap F' \in \mathcal{F}$.

The sets of closure systems $\mathcal{F}$ are called closed sets.

Moreover, any lattice is isomorphic to the lattice of closed sets of a closure system [9]. The simplest closure system representing L is defined on J_L: it is the set system $\{J_x \mid x \in L\}$.

Example 1. Let $\mathcal{F}$ be a closure system on G={a,b,c,d,e}:
$\mathcal{F}_\Sigma$={{},{a}, {b}, {c}, {e}, {ab}, {ad}, {abd}, {abce}, {abcde}}.
The sets of $\mathcal{F}$ ordered by set-inclusion is a lattice denoted by $L_\mathcal{F} = (\mathcal{F}, \subseteq)$. Figure 1 shows the lattice associated to this closure system.

At each closure systems can be associated one or some implicational bases.

An implication $A \rightarrow B$ on G is a pair of subsets A and B of G. A subset $X \subseteq G$ is a model of $A \rightarrow B$ if and only if $A \subseteq X$ implies $B \subseteq X$. A subset X is a model of a set Σ of implications if it is a model for every implication in Σ. $\mathcal{F}_\Sigma$ denotes the set of all models (or Σ-closed sets) of Σ, which is a closure system on G. A set Σ of implications is a basis of a closure system $\mathcal{F}$ if $\mathcal{F} = \mathcal{F}_\Sigma$. Since, closure systems are in bijection with closure operators, the we can associate a closure operator ϕ_Σ as follows :

$$For \quad X \subset G, \quad \phi_\Sigma(X) = X^0 \cup X^1 \cup X^2 \cup .....$$

where $X^0 = X$ and $X^i = X^{i-1} \cup \{B \mid A \rightarrow B \in \Sigma,\ A \subseteq X^i\}$
Note that $\phi_\Sigma(X) = X^i$, where $i \leq |G|$ and $X^i = X^{i+1}$.

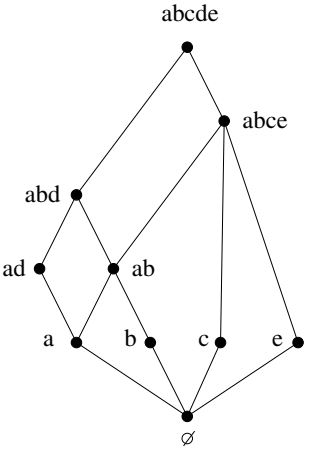


Fig. 1. The lattice $(\mathcal{F}, \subseteq)$ represented by its Hasse diagram

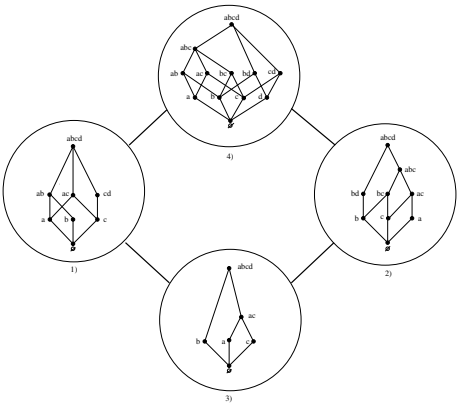


Fig. 2. 1)Lattice corresponding to $\mathcal{F}_1$, 2) Lattice corresponding to $\mathcal{F}_2$, 3) Lattice corresponding to $\mathcal{F}_1 \wedge \mathcal{F}_2$, 4) Lattice corresponding to $\mathcal{F}_1 \vee \mathcal{F}_2$

We say that an implicational base Σ is *direct* or *iteration-free* if for every X $\subseteq$ G, $\phi_\Sigma(X) = X^1$. It means that there is just one passage of Σ to find $\phi_\Sigma(X)$.

Example 2. If we consider closure system of example 1,a possible direct implicational basis of $\mathcal{F}$ is as follow:
$\Sigma_\mathcal{F}$ = { d→a, bd→a, ac→be, bc→ae, ae→bc, be→ac, ce→ab, cd→abe, de→abc}

The set $\mathcal{M}$ of all closure systems on a set G is itself a closure system on the set 2^G. The closure operator on 2^G associated with this closure system is given by :

$$\phi_\mathcal{M}(\mathcal{F}) = \bigcap \{\mathcal{F}' \in \mathcal{M} \mid \mathcal{F} \subseteq \mathcal{F}'\}$$

where $\mathcal{F}$ is an arbitrary family of subsets of G.

The closure system $\mathcal{M}$ ordered under set-inclusion (i.e. $(\mathcal{M}, \subseteq)$ is a lattice whose meet and join operations are given by :

- $\mathcal{F} \wedge \mathcal{F}' = \mathcal{F} \cap \mathcal{F}'$
- $\mathcal{F} \vee \mathcal{F}' = \phi_\mathcal{M}(\mathcal{F} \cup \mathcal{F}') = \{F \cap F' \mid F, F' \in \mathcal{F} \cup \mathcal{F}'\}$.

3 Join of Closure Systems Using Their Implicational Representation

We present here the problem of computing an implicational basis of join of closure systems given by an implicational representation.

Problem 1 (Basis of the Join Operation(BJO)).

Data : A family $\Sigma_{\mathcal{F}_1}, \Sigma_{\mathcal{F}_2}, ..., \Sigma_{\mathcal{F}_k}$ corresponding to implicational basis of the closure systems $\mathcal{F}_1, \mathcal{F}_1, ..., \mathcal{F}_k$ on a finite set G
Question : Compute an implicational basis Σ of $\bigvee_{i=1,k} \mathcal{F}_i$.

3.1 General Case

Authors in [7] shown that can't exist polynomial algorithm in size of the input to resolve problem BJO even if we consider only two closure systems.
We explain this result exhibiting an example.

Example 3. Let Σ_1 and Σ_2 be two minimum implicational basis for $\mathcal{F}_{\Sigma_1}$ and $\mathcal{F}_{\Sigma_2}$.

$$\Sigma_1=\{\emptyset \rightarrow x_0\}$$
$$\Sigma_2=\{x_1...x_n \rightarrow x_0,\ y_1 \rightarrow x_1,.....,\ y_n \rightarrow x_n\}$$

An implicational basis obtained for $\mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$ is the following one:

$$\Sigma =\{z_1....z_n \rightarrow x_0,\ z_i \in \{x_i,y_i\}\ \forall\ \mathrm{i} \in \{1...\mathrm{n}\}\}$$

The fact that this implicational basis is minimum introduces the next proposition:

Proposition 1. *[7] There exists closure systems $\mathcal{F}_1$ and $\mathcal{F}_2$ such that $\Sigma_{\mathcal{F}_1 \vee \mathcal{F}_2}$ has size exponential in the size of $\Sigma_{\mathcal{F}_1}$ and $\Sigma_{\mathcal{F}_2}$.*

Proof. See Example 3.

3.2 Particular Case: Direct Implicational Basis

The previous proposition show us that can't exist a polynomial algorithm in size of the input to compute an implication basis of join of closure systems with their respective basis.

But if we consider a particular class of basis, we show that we can compute an implicational basis for join of closure systems in polynomial time in size of the input. To attempt this goal, it is sufficient to exhibit an algorithm to compute an implicational basis of join of two closure systems. Using this algorithm, an algorithm for a bounded number of closure systems can be directly deduced. We consider then that implicational basis given for closure systems are direct. We now show the following proposition:

Proposition 2. *Problem BJO is polynomial if the given bases are direct bases.*

Proof. see definition 3 and theorem 1.

Definition 3. *Let $\Sigma_1=\{A_1 \rightarrow a_1,...\ A_l \rightarrow a_l\}$ et $\Sigma_2=\{B_1 \rightarrow b_1,...\ B_m \rightarrow b_m\}$ two direct basis associated to two closure systems which are decomposed. We define by:*

$$\Sigma_P = \{\ A_iB_j \rightarrow ((A_iB_j)^{\Sigma_1} \cap (A_iB_j)^{\Sigma_2}) \setminus A_iB_j\ \}$$
for $i \in \{1..l\}$, $j \in \{1..m\}$

With this new implicational basis, we deduce the next theorem:

Theorem 1. $\mathcal{F}_{\Sigma_P} = \mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$.

Proof. We have to show that $\mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2} \subseteq \mathcal{F}_{\Sigma_P}$.
Let F $\in \mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$. Two cases:

- Let F $\in \mathcal{F}_{\Sigma_1} \cup \mathcal{F}_{\Sigma_2}$. We have to show that F is closed under Σ_P. Let A→x $\in \Sigma_P$ and A $\subseteq$ F. Then x $\in$ (A$^{\Sigma_1} \cap A^{\Sigma_2}$)\ A. This implies x $\in$ A$^{\Sigma_1}$ and so that x $\in$ F.
- Let F $\notin \mathcal{F}_{\Sigma_1} \cup \mathcal{F}_{\Sigma_2}$. We have F $= F_1 \cap F_2$ with $F_1 \in \mathcal{F}_{\Sigma_1}$, $F_2 \in \mathcal{F}_{\Sigma_2}$. Let A→x $\in \Sigma_P$ and A $\subseteq$ F, by construction of Σ_P, we obtain x $\in$ (A$^{\Sigma_1} \cap A^{\Sigma_2}$)\ A. This implies x $\in$ A$^{\Sigma_1}$ and x $\in$ A$^{\Sigma_2}$ and so that x $\in$ F because A $\subseteq F_1$ and A $\subseteq F_2$.

Now, we have to prove $\mathcal{F}_{\Sigma_P} \subseteq \mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$. By this way, we have to proof that F $\in \mathcal{F}_{\Sigma_P} \backslash (\mathcal{F}_{\Sigma_1} \cup \mathcal{F}_{\Sigma_2})$ implie F $\in \mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$ i.e F $=$ F$^{\Sigma_1} \cap$ F$^{\Sigma_2}$.

Suppose F $\neq$ F$^{\Sigma_1} \cap F^{\Sigma_2}$. Then $\exists$ x$\notin$ F with x $\in$ F$_{\Sigma_1}$ and x $\in$ F$_{\Sigma_2}$.by the fact that Σ_1 and Σ_2 are direct bases we have , F$^{\Sigma_1}$ = F $\cup \bigcup\{\phi(A) | A \rightarrow C \in \Sigma_1$ *and* $A \subseteq F\}$ and F$^{\Sigma_2}$ = F $\cup \bigcup\{\phi(B) | B \rightarrow D \in \Sigma_2$ *and* $B \subseteq F\}$. So there exists A→C $\in \Sigma_1$ with x $\in$ C and B→D $\in \Sigma_2$ with x $\in$ D. By definition 3 , AB → x $\in \Sigma_P$ et AB $\subseteq$ F. So there is a contradiction since F is a closed set of $\mathcal{F}_{\Sigma_P}$. As a conclusion, $\mathcal{F}_{\Sigma_P} \subseteq \mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$. □

With the previous theorem, Σ_P represents a implicational basis $\mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$. Then, we can compute an implicational base for $\mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$ with Σ_1 and Σ_2. Moreover, this computing is polynomial in both side of Σ_1 and Σ_2.

Theorem 2. *Let Σ_1, Σ_2 be two direct bases. Then compute a implicational basis for $\mathcal{F}_{\Sigma_1 \vee \Sigma_2}$ can be made in polynomial time in size of the input.*

Algorithm 1. implicational basis of join of closure systems

Input: Σ_1, Σ_2, two direct basis of $\mathcal{F}_1$ and $\mathcal{F}_2$
Output: an implicational basis of $\mathcal{F}_1 \vee \mathcal{F}_2$
begin
  $\Sigma_P = \emptyset$
  foreach *rule* $A \rightarrow C$ *in* Σ_1 **do**
    foreach *rule* $B \rightarrow D$ *in* Σ_2 **do**
      compute Ext= $(AB)^{\Sigma_1} \cap (AB)^{\Sigma_2}) \backslash AB$
      if $Ext \neq \emptyset$ **then**
        $\Sigma_P = \Sigma_P \cup AB \rightarrow$ Ext.
  return Σ_P
end

Example 4. Consider following Σ_1 and Σ_2 which are direct implicational basis of two closure systems $\mathcal{F}_{\Sigma_1}$ and $\mathcal{F}_{\Sigma_2}$.

$$\Sigma_1 = \{ \quad \Sigma_2 = \{$$

a→c	b→d
ab→c	ab→d
ce→d	de→c
ad→c	bc→d
bce→d	ade→c
abd→c	abc→d
ae→cd	be→cd
abe→cd	abe→cd

$\}$ $\}$

We compute $\mathcal{F}_{\Sigma_1} \vee \mathcal{F}_{\Sigma_2}$ dropping duplicated rules and redundant rules:

$$\Sigma_P = \{$$

ade→c
abe→cd
bce→d

$\}$

This result can be generalized for a bounded number, by a constant k, of closure systems. We can apply the same process viewed for two closure systems. The algorithm is always polynomial since the number of implicational rules generated remains polynomial due to the constant number of implicational basis taken as input.

4 Conclusion

In this paper, we studied the problem of computing an implicational basis of join of bounded number closure systems represented by their implicational basis. We produces a polynomial algorithm in the particular case where implicational basis are direct. This problem remains open in general case if we consider both size of input and size of the output.

Further work is to look if these results can be applied to distributed implicational rules discovery.

References

1. Guigues, J.-L., Duquenne, V.: Famille minimale d'implications informatives résultant d'un tableau de donneés binaires, Mathemátiques et sciences humaines, 24 (1986)
2. Eiter, T., Gottlob, G.: Hypergraph transversal computation and related problems in logic and AI. In: European Conference on Logics in Artificial Intelligence (JELIA 2002), pp. 549–564 (2002)
3. Ganter, B.: Finding closed sets under symmetry, FB4-Preprint 1307, TH Darmstadt (1990)

4. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations. Springer, Heidelberg (1999)
5. Mannila, H., Räihä, K.J.: On the complexity of inferring functional dependencies. Discrete Applied Mathematics 40(2), 237–243 (1992)
6. Eiter, T., Ibaraki, T., Makino, K.: Computing Intersections of Horn Theories for Reasoning with Models. Artificial Intelligence 110(1), 57–101 (1999)
7. Eiter, T., Ibaraki, T., Makino, K.: Disjunctions of Horn Theories and their Cores, Rutcor Research Report (1998)
8. Kautz, H., Selman, B.: Knowledge Compilation and Theory Approximation. Journal of the ACM 43(2), 193–224 (1996)
9. Birkhoff, G., Frink, O.: Representations of lattices of sets. Transactions of the American Mathematical Society 64, 299–316 (1948)

G^1 Blending B-Spline Surfaces and Optimization

Bachir Belkhatir[1], Driss Sbibih[1,⋆], and Ahmeh Zidna[2]

[1] Université Mohammed I, Ecole Supérieure de Technologie ,
Laboratoire MATSI, Oujda, Maroc
bachirbelkhatir@yahoo.fr,
sbibih@yahoo.fr
[2] Laboratoire de l'Informatique Théorique et Appliquée,
UFR Scientifique MIM Université Paul Verlaine-Metz,
Ile du Saulcy, 57045 Metz, France
zidna@univ-metz.fr

Abstract. Recently, some results on G^1 continuity conditions for two adjacent B-spline surfaces such as bicubic or biquartic B-spline surfaces have been developed in the literature. However, the blending and the optimization problems related to these surfaces were not studied. Then, we give in this paper a method which allows to solve a G^1 blending problem of two B-spline surfaces and an algorithm for finding optimal surfaces.

Keywords: B-spline, Blending surface, optimization.

1 Introduction

The construction of blending parametric surfaces, such as Bézier patches and B-spline surfaces, is a functional problem in computer graphics, animation, geometric modeling, CAD/CAM and reverse engineering. It consists in constructing a blending surface that smoothing joins on more given surfaces. A blending surface is widely used for functional or esthetic reasons in geometric design. The study of the geometric continuity between two Bézier patches received a considerable importance in the field of CAGD (see Liu and Hoschek, 1989, Liu, 1990), Du and Schmitt, 1990, Zheng et al., 1995, Ye et al., 1996). But a discussion of the geometric continuity conditions between two B-spline surfaces was seldom seen before 1995. Shi et al. (2002) have presented the necessary and the sufficient conditions of G^1 continuity between two biquartic B-spline surfaces with single interior knots. The problem for the G^1 construction blending surface can be formulated as follows. Let $A(u, v)$ and $B(u, v)$ be two C^1 B-spline surfaces. We assume that these two surfaces have the same degree and knot vector in the v-direction. We want to build a B-spline surface $C(u, v)$ which satisfies the positional continuity of the surfaces A, B and the blending surface C, i.e. $C(0, v) = A(0, v)$ and $C(1, v) = B(1, v)$. We also want that the blending surface satisfies G^1 continuity with the given surfaces A and B. This last constraint

⋆ This work is supported by the project AI MA/08/182.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 458–467, 2008.

gives rise to four parameter functions:

$$\alpha_0(v) = \alpha_0, \quad \alpha_1(v) = \alpha_1, \quad \beta_0(v) = \sum_{i=0}^{\bar{m}+1} \beta_i^0 N_{i,1}(v) \quad \text{and} \quad \beta_1(v) = \sum_{i=0}^{\bar{m}+1} \beta_i^1 N_{i,1}(v)$$

such that

$$\begin{cases} \frac{\partial C}{\partial u}(0,v) = \alpha_0 \frac{\partial A}{\partial u}(0,v) + \beta_0 \frac{\partial A}{\partial v}(0,v), \\ \\ \frac{\partial C}{\partial u}(1,v) = \alpha_1(v) \frac{\partial B}{\partial u}(1,v) + \beta_1(v) \frac{\partial B}{\partial v}(1,v). \end{cases}$$

The problem that arises here is how to choose the parameter functions in order to obtain an appropriate blending surface in unique way. Then, the definition of the blending surface shape depends on the choice of the arbitrary functions mentioned above. When they are fixed, one can take the solution of some boundary-value problem, which is directly related with a functional to be minimized. With the free parameters α_0, α_1 and β_i^k where $k \in \{0,1\}$ and $i \in \{0,\ldots,\bar{m}+1\}$, this problem admits some solutions. Our optimization process tries to find a B-spline surface for which the quality of the surface is optimum. In this paper, we use the numerical optimization method to deal with geometric constraints for solving this problem. A numerical example is given for illustrating the theoretical results.

2 B-Spline Surface Review

In this section the representation of B-spline patches and some of their geometric properties are briefly recalled. For more detailed description, the reader can consult [9]. A tensor product B-spline is defined by

$$A(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} a_{i,j} N_{i,p}(u) N_{j,q}(v), \tag{1}$$

where a_{ij} are the control points in the space $\mathbb{R}^3$ and $N_{i,k}$ ($k = p$ or q) are the B-spline basis functions of degree k defined respectively on the non periodic knot vectors:

$$U = [\underbrace{0,\ldots,0}_{p+1}, \underbrace{u_l,\ldots,u_l}_{k_l}, \ldots, \underbrace{u_{\bar{n}},\ldots,u_{\bar{n}}}_{k_{\bar{n}}}, \underbrace{1,\ldots,1}_{p+1}],\ 1 \le k_l < p,$$

$$V = [\underbrace{0,\ldots,0}_{q+1}, \underbrace{v_l,\ldots,v_l}_{k_l}, \ldots, \underbrace{v_{\bar{m}},\ldots,v_{\bar{m}}}_{k_{\bar{m}}}, \underbrace{1,\ldots,1}_{q+1}],\ 1 \le k_l < q,$$

where k_l, $1 \le l \le \bar{n}$ (resp. $1 \le l \le \bar{m}$) is the multiplicity of the interior knot u_l (resp. v_l). Hence, $n = \sum_{l=1}^{\bar{n}} k_l + p$ and $m = \sum_{l=1}^{\bar{m}} k_l + q$.

Therefore, the knot vectors U and V can be written in the form

$$U = [\bar{u}_0, \bar{u}_1, \ldots, \bar{u}_{n+p+1}] \text{ and } V = [\bar{v}_0, \bar{v}_1, \ldots, \bar{v}_{m+q+1}].$$

Now, let us consider another B-spline surface $B(u,v)$ defined by

$$B(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} b_{i,j} N_{i,p}(u) N_{j,q}(v). \tag{2}$$

According to the the properties of B-splines, we get

$$\frac{\partial A}{\partial u}(u,v) \mid_{u=0} = \sum_{j=0}^{m} \frac{p}{u_1}(a_{j,1} - a_{j,0}) N_{j,q}(v),$$

and

$$\frac{\partial A}{\partial v}(u,v) \mid_{u=0} = \sum_{j=0}^{m-1} \frac{q}{\bar{v}_{j+q+1} - \bar{v}_{j+1}}(a_{j+1,0} - a_{j,0}) N_{j,q-1}(v),$$

where $v \in [0,1]$, and $N_{j,q-1}$ are the B-spline basis functions of degree $q-1$ defined on the knot vector

$$\bar{V} = [\underbrace{0,\ldots,0}_{q}, \underbrace{v_l,\ldots,v_l}_{k_l},\ldots,\underbrace{v_{\bar{m}},\ldots,v_{\bar{m}}}_{k_{\bar{m}}}, \underbrace{1,\ldots,1}_{q}].$$

Similarly, for the surface B, we have

$$\frac{\partial B}{\partial u}(u,v) \mid_{u=0} = \sum_{i=0}^{m} \frac{p}{u_1}(b_{j,1} - b_{j,0}) N_{j,q}(v),$$

and

$$\frac{\partial B}{\partial v}(u,v) \mid_{u=0} = \sum_{j=0}^{m-1} \frac{q}{\bar{v}_{j+q+1} - \bar{v}_{j+1}}(b_{j+1,0} - b_{j,0}) N_{j,q-1}(v).$$

Moreover, the cross-boundary tangent vector for this surface at $u = 1$ is given by

$$\frac{\partial B}{\partial u}(u,v) \mid_{u=1} = \sum_{j=0}^{m} \frac{p}{\bar{u}_{2p}}(b_{j,n} - b_{j,n-1}) N_{j,q}(v),$$

and

$$\frac{\partial B}{\partial v}(u,v) \mid_{u=1} = \sum_{j=0}^{m-1} \frac{q}{\bar{v}_{j+q+1} - \bar{v}_{j+1}}(b_{j+1,n} - b_{j,n}) N_{j,q-1}(v).$$

In the next section, we will study the G^1 continuity conditions between the surfaces (1) and (2).

3 Sufficient Conditions of G^1continuity of Two B-Spline Patches

According to [12], the necessary and sufficient conditions for the two surfaces (1) and (2) joining G^0-continuously along a common boundary curve $R(v)$ are such that

$$R(v) = A(0,v) = B(0,v), \qquad \forall\, v \in [0,1], \tag{3}$$

and these surfaces are G^1-continuously jointed along the curve $R(v)$ if and only if there exist two scalar functions α and β defined on $[0,1]$ and satisfying the following equation

$$\frac{\partial B}{\partial u}(0,v) = \alpha(v)\frac{\partial A}{\partial u}(0,v) + \beta(v)\frac{\partial A}{\partial v}(0,v) \qquad \forall v \in [0,1]. \tag{4}$$

In what follows, we take $\alpha(v) = \alpha$ and $\beta(v) = \sum_{i=0}^{\bar{m}+1} \beta_i N_{i,1}\ \forall v \in [0,1]$.

Denote by V_l the interval $[v_{l-1}, v_l]$, $1 \le l \le \bar{m}+1$, with $v_0 = 0, v_{\bar{m}+1} = 1$.

For each $v \in V_l$, we denote by $A_l(u,v)$ and $B_l(s,v)$ the l^{th} patches of $A(u,v)$ and $B(s,v)$ respectively, and $R_l(v) = R(v)$.

According to [12], the surfaces $A_l(u,v)$ and $B_l(u,v)$ can be regarded as Bézier surface patches. By using the knot refinement process (see [9]), we can write

$$\frac{\partial B}{\partial u}\rfloor_l(0,v) = \sum_{j=0}^{q} b_j^l B_{j,q}(t), \qquad \frac{\partial A}{\partial u}\rfloor_l(0,v) = \sum_{j=0}^{q} a_j^l B_{j,q}(t)$$

and

$$\frac{\partial A}{\partial v}\rfloor_l(0,v) = \sum_{j=0}^{q-1} \bar{a}_j^l B_{j,q-1}(t).$$

Theorem 1. *The two surfaces $A(u,v)$ and $B(u,v)$ defined by (1) and (2) respectively are G^1 continuous along their common boundary curve $R(v)$ if the following conditions are satisfied*

1. $a_{i,0} = b_{i,0}, \qquad i = 0,1,\ldots,m.$
2. $b_j^l = \alpha a_j^l + \beta_{l-1}(\frac{q-j}{q})\bar{a}_j^l + \beta_l \frac{j}{q}\bar{a}_{j-1}^l$, *where* $a_{-1}^l = a_q^l = 0$, $l = 1,2,\ldots,\bar{m}+1$ *and* $j = 0,1,\ldots,q$.
3. $b_q^l = b_0^{l+1}, \bar{a}_0^{l+1} = \bar{a}_{q-1}^l$ *and* $a_q^l = a_0^{l+1}$, $l = 1,2,\ldots,\bar{m}$.

Proof. According to the expressions of the surfaces A and B, (3) is equivalent to

$$a_{i,0} = b_{i,0}, \qquad \forall\ i = 0,1,\ldots,m. \tag{5}$$

Then, the surfaces A and B are G^0 continuous along their common boundary curve $R(v)$.

On the other hand, for $v \in [v_{l-1}, v_l]$, $l = 1,2,...,\bar{m}+1$, we have

$$\alpha(v) = \alpha \ \text{ and } \ \beta(v) = \beta_{l-1}(1-t) + \beta_l t, \ \text{ where } \ t = \frac{v - v_{l-1}}{v_l - v_{l-1}}.$$

Then, as

$$tB_{j,q-1}(t) = \frac{(j+1)}{q}B_{j+1,q}(t) \quad \text{and} \quad (1-t)B_{j,q-1}(t) = \frac{q-j}{q}B_{j,q}(t), \tag{6}$$

we deduce from the condition (2) in the above theorem that $\forall t \in [0,1]$ we have

$$\sum_{j=0}^{q} b_j^l B_{j,q}(t) = \alpha \sum_{j=0}^{q} \bar{a}_j^l B_{j,q}(t) + (\beta_{l-1} t + \beta_l (1-t)) \sum_{j=0}^{q-1} a_j^l B_{j,q-1}(t). \tag{7}$$

Therefore, $\forall v \in V_l$ we get

$$\frac{\partial B}{\partial u}\rfloor_l(0,v) = \alpha \frac{\partial A}{\partial u}\rfloor_l(0,v) + \beta_l(v) \frac{\partial A}{\partial v}\rfloor_l(0,v). \tag{8}$$

Finally, the condition (3), i.e. $b_q^l = b_0^{l+1}, a_0^{l+1} = a_q^l$ and $\bar{a}_{q-1}^l = \bar{a}_0^{l+1}$, ensures the continuity of $\frac{\partial B}{\partial u}(0,v)$, $\frac{\partial A}{\partial u}(0,v)$ and $\frac{\partial A}{\partial v}(0,v)$ at the interior knots v_l, then the equation (5) is satisfied.

4 Construction of a G^1 Blending Patch of Two B-Splines

Let $A(u,v)$ and $B(u,v)$ be two C^1 B-spline surfaces as mentionned above. For all $v \in [0,1]$, we want to construct a B-spline surface $C(u,v)$ which satisfies the following constraints:

$$\begin{cases} C(0,v) = A(0,v), \\ C(1,v) = B(1,v), \\ \\ \frac{\partial C}{\partial u}(0,v) = \alpha_0 \frac{\partial A}{\partial u}(0,v) + \beta_0(v) \frac{\partial A}{\partial v}(0,v), \\ \\ \frac{\partial C}{\partial u}(1,v) = \alpha_1 \frac{\partial B}{\partial u}(1,v) + \beta_1(v) \frac{\partial B}{\partial v}(1,v), \end{cases} \tag{9}$$

where $\alpha_0, \alpha_1 \in \mathbb{R}$, $\beta_0(v) = \sum_{i=0}^{\bar{m}+1} \beta_i^0(v) N_{i,1}(v)$ and $\beta_1(v) = \sum_{i=0}^{\bar{m}+1} \beta_i^1(v) N_{i,1}(v)$.

One way to define the blending B-spline surface is to use the tensor product by putting

$$C(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{3} c_{ij} B_{j,3}(u) N_{i,q}(v), \tag{10}$$

where $B_{j,3}(u)$ is the classical Bernstein basis polynomials of degree 3 defined by

$$B_{j,3}(u) = \frac{3!}{j!(3-j)!} u^j (1-u)^{3-j}, \quad j = 0, \ldots, 3. \tag{11}$$

In this case, it is simple to verify that for $v \in V_l$, the constraints (9) become

$$\begin{cases} \frac{\partial C}{\partial u}\rfloor_l(0,v) = \alpha_0 \frac{\partial A}{\partial u}\rfloor_l(0,v) + \beta_{0,l}(v) \frac{\partial A}{\partial v}\rfloor_l(0,v), \\ \\ \frac{\partial C}{\partial u}\rfloor_l(1,v) = \alpha_1 \frac{\partial B}{\partial u}\rfloor_l(1,v) + \beta_{1,l}(v) \frac{\partial B}{\partial v}\rfloor_l(1,v), \end{cases} \tag{12}$$

where $\beta_{0,l}(v) = \beta_{l-1}^0(1-t) + \beta_l^0 t$, $\beta_{1,l}(v) = \beta_{l-1}^1(1-t) + \beta_l^1 t$, with $t = \frac{v - v_{l-1}}{v_l - v_{l-1}}$.

4.1 Solution for Two Bicubic B-Spline Surfaces

For the sake of simplicity, we restrict our study in this subsection to the case of two uniform bicubic G^1 blending B-spline surfaces, that is $p = q = 3$, with $n = m \geq 8$. For other cases, one can obtain similar results using this method.

In this case, equations (1), (2) and (10) can be written as follows

$$A(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{m} a_{ij} N_{j,3}(u) N_{i,3}(v), \tag{13}$$

$$B(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{m} b_{ij} N_{j,3}(u) N_{i,3}(v), \tag{14}$$

$$C(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{3} c_{i,j} B_{j,3}(u) N_{i3}(v), \tag{15}$$

where $U = V = \{0,0,0,0,v_1,v_2,\ldots,v_{\bar{m}},1,1,1,1\}, v_0 = 0, v_{\bar{m}+1} = 1$ and $v_{l+1} - v_l = \frac{1}{m-2}, l = 0,\ldots,\bar{m}+1$.

The cross-boundary tangent vector for the surfaces A, B and C are

$$\begin{cases} \frac{\partial A}{\partial u}(0,v) = \sum_{i=0}^{m} \frac{3}{h}(a_{i,1} - a_{i,0}) N_{i,3}(v) = \sum_{i=0}^{m} a_i N_{i,3}(v), \text{ with } h = \frac{1}{m-2}, \\ \frac{\partial A}{\partial v}(0,v) = \sum_{i=0}^{m-1} \frac{3}{\bar{v}_{i+4} - \bar{v}_{i+1}}(a_{i+1,0} - a_{i,0}) N_{i,2}(v) = \sum_{i=0}^{m-1} \bar{a}_i N_{i,2}(v), \\ \frac{\partial B}{\partial u}(1,v) = \sum_{i=0}^{m} \frac{3}{\bar{v}_6}(b_{i,m} - b_{i,m-1}) N_{i,3}(v) = \sum_{i=0}^{m} b_i N_{i,3}(v), \\ \frac{\partial B}{\partial v}(1,v) = \sum_{i=0}^{m-1} \frac{3}{\bar{v}_{i+4} - \bar{v}_{i+1}}(b_{i+1,m} - b_{i,m}) N_{i,2}(v) = \sum_{i=0}^{m-1} \bar{b}_i N_{i,2}(v), \\ \frac{\partial C}{\partial u}(0,v) = \sum_{i=0}^{m} 3(c_{i,1} - c_{i,0}) N_{i,3}(v) = \sum_{i=0}^{m} c_i^0 N_{i,3}(v), \\ \frac{\partial C}{\partial u}(1,v) = \sum_{i=0}^{m} 3(c_{i,3} - c_{i,2}) N_{i,3}(v) = \sum_{i=0}^{m} c_i^1 N_{i,3}(v), \end{cases} \tag{16}$$

where

$$\begin{cases} a_i = \frac{3}{h}(a_{i,1} - a_{i,0}), & \bar{a}_i = \frac{3}{\bar{v}_{i+4} - \bar{v}_{i+1}}(a_{i+1,0} - a_{i,0}), \\ b_i = \frac{3}{\bar{v}_6}(b_{i,3} - b_{i,2}), & \bar{b}_i = \frac{3}{\bar{v}_{i+4} - \bar{v}_{i+1}}(b_{i+1,3} - b_{i,3}), \\ c_i^0 = 3(c_{i,1} - c_{i,0}), & c_i^1 = 3(c_{i,3} - c_{i,2}). \end{cases} \tag{17}$$

The four steps of the algorithm for constructing the G^1 B-spline blending surface are as follows.

Step 1. *Determination of the control points* $\{c_{i,0}, c_{i,3}\}$.
From the G^0 continuity, we deduce that $c_{i,0} = a_{i,0}$ and $c_{i,3} = b_{i,m}$, for all $i = 0,\ldots,m$,

Step 2. *Knot refinement.*
Using a knot refinement technique, see [9], we can write

$$\sum_{i=0}^{m} a_i N_{i,3}(v) = \sum_{i=0}^{3(m-2)} \hat{a}_i \hat{N}_{i,3}(v), \qquad \sum_{i=0}^{m} b_i N_{i,3}(v) = \sum_{i=0}^{3(m-2)} \hat{b}_i \hat{N}_{i,3}(v), \quad (18)$$

$$\sum_{i=0}^{m} c_i^0 N_{i,3}(v) = \sum_{i=0}^{3(m-2)} \hat{c}_i^0 \hat{N}_{i,3}(v), \qquad \sum_{i=0}^{m} c_i^1 N_{i,3}(v) = \sum_{i=0}^{3(m-2)} \hat{c}_i^1 \hat{N}_{i,3}(v), \quad (19)$$

where $\hat{N}_{i,3}$ are the basis B-splines of degree 3 defined on the knot vector

$$\hat{V}_2 = [0,0,0,0,v_1,v_1,v_1,...,v_{\bar{m}},v_{\bar{m}},1,1,1,1].$$

Similarly we have

$$\sum_{i=0}^{m-1} \bar{a}_i N_{i,2}(v) = \sum_{i=0}^{2(m-2)} \hat{\bar{a}}_i \hat{N}_{i,2}(v), \quad \sum_{i=0}^{m} \bar{b}_i N_{i,2}(v) = \sum_{i=0}^{2(m-2)} \hat{\bar{b}}_i \hat{N}_{i,2}(v), \quad (20)$$

where $\hat{N}_{i,2}$ are the basis B-splines of degree 2 defined on the knot vector

$$\hat{V}_1 = [0,0,0,v_1,v_1,...,v_{\bar{m}},v_{\bar{m}},1,1,1].$$

The coefficients $\{a_i\}_{0\le i\le(3m-6)}$ and $\{\hat{\bar{a}}_i\}_{0\le i\le(2m-4)}$ are determined as follows:

$$\begin{cases}
\hat{a}_0 = a_0, \quad \hat{a}_1 = a_1, \quad \hat{a}_2 = \frac{1}{2}(a_1+a_2), \quad \hat{a}_3 = \frac{1}{12}(3a_1+7a_2+2a_3),\\
\hat{a}_{3i} = \frac{1}{6}(a_i+4a_{i+1}+a_{i+2}), \quad i=2,...,m-4,\\
\hat{a}_{3i+1} = \frac{1}{3}(2a_{i+1}+a_{i+2}), \quad \hat{a}_{3i+2} = \frac{1}{3}(a_{i+1}+2a_{i+2}), \quad i=1,...,m-4,\\
\hat{a}_{3(m-3)} = \frac{1}{12}(2a_{m-3}+7a_{m-2}+3a_{m-1}),\\
\hat{a}_{3(m-3)+1} = \frac{1}{2}(a_{m-2}+a_{m-1}), \quad \hat{a}_{3(m-3)+2} = a_{m-1}, \quad \hat{a}_{3(m-2)} = a_m,\\
\hat{\bar{a}}_0 = \bar{a}_0, \quad \hat{\bar{a}}_1 = \frac{1}{2}\bar{a}_1, \quad \hat{\bar{a}}_2 = \frac{1}{12}(3\bar{a}_1+2\bar{a}_2),\\
\hat{\bar{a}}_{2i+1} = \frac{1}{3}\bar{a}_{i+1}, \ i=1,...,m-4, \quad \hat{\bar{a}}_{2i} = \frac{1}{6}(\bar{a}_i+\bar{a}_{i+1}), \ i=2,...,m-4,\\
\hat{\bar{a}}_{2(m-3)} = \frac{1}{12}(2\bar{a}_{m-3}+3\bar{a}_{m-2}), \quad \hat{\bar{a}}_{2(m-3)+1} = \frac{1}{2}\bar{a}_{m-2}, \quad \hat{\bar{a}}_{2(m-2)} = \bar{a}_{m-1}.
\end{cases} \quad (21)$$

The coefficients $\{\hat{b}_i\}, \{\hat{c}_i^1\}, \{\hat{c}_i^0\}$ and $\{\hat{\bar{b}}_i\}$ can be computed in a similar way.

Step 3. G^1 *conditions between* $C(u,v), A(u,v)$ *and* $C(u,v), B(u,v)$. From (10), the piecewise Bézier control points $\{\hat{c}^0_{3l+i}\}$ and $\{\hat{c}^1_{3l+i}\}$, $i = 0,1,2,3$, and $l =$

$1, ..., m-2$ can be obtained as follows:

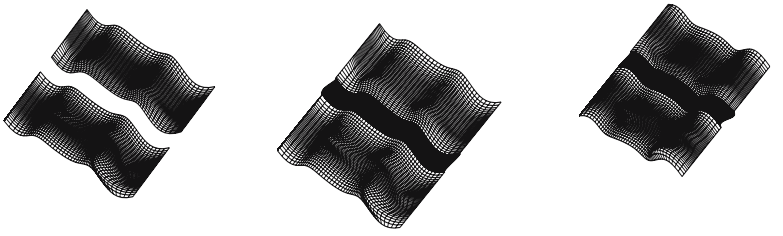
$$\begin{cases} \hat{c}^0_{3l} = \alpha_1 \hat{a}_{3l} + \beta^0_{l-1} \hat{\bar{a}}_{2l}, \\ \hat{c}^0_{3l+1} = \alpha_1 \hat{a}_{3l+1} + \frac{2}{3}\beta^0_{l-1} \hat{\bar{a}}_{2l+1} + \frac{1}{3}\beta^0_l \hat{\bar{a}}_{2l}, \\ \hat{c}^0_{3l+2} = \alpha_1 \hat{a}_{3l+2} + \frac{1}{3}\beta^0_{l-1} \hat{\bar{a}}_{2l+2} + \frac{2}{3}\beta^0_l \hat{\bar{a}}_{2l+2}, \\ \hat{c}^0_{3l+3} = \alpha_1 \hat{a}_{3l+3} + \beta^0_l \hat{\bar{a}}_{2l+2}, \\ \hat{c}^1_{3l} = \alpha_2 \hat{b}_{3l} + \beta^1_{l-1} \hat{\bar{b}}_{2l}, \\ \hat{c}^1_{3l+1} = \alpha_2 \hat{b}_{3l+1} + \frac{2}{3}\beta^1_{l-1} \hat{\bar{b}}_{2l+1} + \frac{1}{3}\beta^1_l \hat{\bar{b}}_{2l}, \\ \hat{c}^1_{3l+2} = \alpha_2 \hat{b}_{3l+2} + \frac{1}{3}\beta^1_{l-1} \hat{\bar{b}}_{2l+2} + \frac{2}{3}\beta^1_l \hat{\bar{b}}_{2l+2}, \\ \hat{c}^1_{3l+3} = \alpha_2 \hat{b}_{3l+3} + \beta^1_l \hat{\bar{b}}_{2l+2}. \end{cases} \tag{22}$$

Step 4. *Computation of the control points* $\{c_{i,j}, i = 0, ..., m \text{ and } j = 1, 2\}$.

After determining $\{a_i\}, \{\bar{a}_i\}, \{b_i\}, \{\bar{b}_i\}, \{c^0_i\}$ and $\{c^1_i\}, i = 0, ..., m$, introduced in (17), the coefficients $\{c_{i,j}, i = 0, ..., m \text{ and } j = 1, 2\}$ are computed by using Steep 2.

4.2 Numerical Example

For $n = m = 8$ and $U = V = [0, 0, 0, 0, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, 1, 1, 1, 1]$, we show in Figure 1 (a) the graphs of their corresponding B-spline surfaces of degree 3. In Figure 1(b) and Figure 1 (c), we give their G^1 blending surfaces by using two choices of parameter functions.



(a) (b) (c)

Fig. 1. G^1 blending surfaces between two bicubic B-spline surfaces. For Fig.1(b): $\alpha_0 = -1, \alpha_1 = 1, \beta_0 = \beta_1 = [5, 5, 5, 5, 0.05, 1, 0.05]$. For Fig.1(c): $\alpha_0 = -4, \alpha_1 = 1, \beta_0 = [0.05, 0.05, 0.05, 0.05, 0.05, 0.06, 0.05], \beta_1 = [2, 1, 3, 5, 0.05, 1, 1]$.

5 Optimization of Blending B-Spline Surfaces

As the parameters α_1, α_2 and β^k_l, $k \in \{1, 2\}$ and $l \in \{0, \ldots, \bar{m} + 1\}$, are free, the problem of blending B-spline surfaces admits several solutions. Then, we propose, in this section, an optimization process for finding a blending B-spline surface such that the quality of the global surface is optimum. More precisely,

we want to build a blending surface from above problem which will be more symmetric in the sense that for all $i = 0, ..., m-2$, $\|c_{i,1} - c_{i,0}\|_2$ or $\|c_{i,3} - c_{i,2}\|_2$ is minimal and $\|c_{i,1} - c_{i,0}\|_2 = \|c_{i,3} - c_{i,2}\|_2$, for all $i = 0, ..., m$.

The algorithm that allows us to build this optimal surface is based on the two following propositions which can easily be proved.

Proposition 1. *For $i = 0, ..., m-2$, we have $c_{i,1} - c_{i,0} = u_i^0 + \beta_i^0 v_i^0$ and $c_{i,3} - c_{i,2} = u_i^1 + \beta_i^1 v_i^1$ where u_i^0, u_i^1, v_i^0 and v_i^1 are given vectors.*

Proposition 2. *Let x and y be two vectors in $\mathbb{R}^3$. If $y \neq 0$ then there exists a real $\bar{\beta} = \frac{-<x,y>}{\|y\|_2^2}$ which minimizes the expression $\|x + \beta y\|_2$.*

Description of the algorithm
Output $\{\beta_i^0, \beta_i^1, i = 0, ..., m-2\}$
For i=0 : m-2
  Find
    $\epsilon_i^0(\beta^0) \longleftarrow \min_{\beta \in \mathbb{R}} \|u_i^0 + \beta v_i^0\|_2$ **and** $\epsilon_i^1(\beta^1) \longleftarrow \min_{\beta \in \mathbb{R}} \|u_i^1 + \beta v_i^1\|_2$
  If $\epsilon_i^0(\beta^0) \leq \epsilon_i^1(\beta^1)$ **then**
    $\beta_i^1 \longleftarrow \beta^1$
    Solve $\|u_i^0 + \beta v_i^0\|_2 = \epsilon_i^1(\beta^1)$
    $\beta_i^0 \longleftarrow \beta$
  Else
    $\beta_i^0 \longleftarrow \beta^0$
    Solve $\|u_i^1 + \beta v_i^1\|_2 = \epsilon_i^0(\beta^0)$
    $\beta_i^1 \longleftarrow \beta$
  End
End

Now, if we reconsider the example introduced above, then with the choice $\alpha_1 = \alpha_2 = -1$, we obtain the following G^1 blending optimal surface.

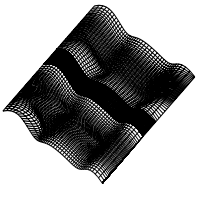


Fig. 2. A G^1 Blending optimal surface

6 Conclusion and Perspectives

In this work we have presented a method for constructing a G^1 B- spline surface which blends two given B-spline surfaces. The blending surface is optimized for an optimum geometric quality surface. In the next work, we will exploit the free parameters for minimizing one of functionals $J_1(C) = \int_\Omega \|\frac{\partial C}{\partial u} \wedge \frac{\partial C}{\partial v}\|_2 dudv$ and $J_2(C) = \int_\Omega H(u,v)^2 dudv$ where $\Omega = [0,1] \times [0,1]$ and $H(u,v)$ denote the mean curvature of the blending surface C.

References

1. Bartels, R., Beatty, J., Barsky, B.: An Introduction on Splines for Use in Computer Graphics and Geometric Modeling. Morgan Kaufman, Los Altos (1987)
2. Che, X., Liang, X., Li, Q.: G^1 continuity for adjacent NURBS surfaces. Computer Aided Geometric Design 22(4), 285–298 (2005)
3. Coons, S.: Surface patches and B-spline curves. In: Barnhill, R., Riesenfeld, R. (eds.) Computer Aided Geometric Design, pp. 1–16. Academic Press, London (1974)
4. Du, W.-H., Schmitt, F.: On the G^1 continuity of piecewise Bézier surfaces: a review with new results. Computer Aided Design 22(9), 556–573 (1990)
5. Farin, G., Hoschek, J., Kim, M.S.: Handbook of computer aided geometric design. Computer-Aided Design 37 (2005)
6. Ferguson, D.R.: Construction of curves and surfaces using numerical optimization techniques. Computer-Aided Design 18, 15–21 (1986)
7. Ferguson, D.R., Frank, P.D., Jones, A.K.: Surface Sharp control using constrained optimization on the B-spline representation. Computer Aided Geometric Design 5, 87–103 (1988)
8. Manning, J.: Continuity conditions for spline curves. The Computer J. 17(2), 181–186 (1974)
9. Piegl, L., Tiller, W.: The NURBS Book, 2nd edn. Springer, Heidelberg (1997)
10. Shi, X., Wang, T., Wu, P., Liu, F.: Reconstruction of convergent G^1 smooth B-spline surfaces. Computer-Aided Geometric Design 21, 893–913 (2004)
11. Szilvasi-Nagy, M.: Shaping and fairning of tubular B-spline surfaces. Computer Aided Design 14, 699–706 (1997)
12. Zheng, J., Wang, G., Liang, Y.: GC^n continuity conditions for adjacent Bézier patches and their constructions. Computer Aided Geometric Design 13, 521–548 (1995)

A Delineation Algorithm for Particle Images Online

Weixing Wang, Chunzhi Wang, Yanzhong Hu, and Wei Liu

Collage of Computer Science and Technology, Hubei University of Technology, Wuhan city, Hubei province, China
Znn525d@qq.com

Abstract. The geometry properties of a particle are very important pieces of information for production optimization in many industrial applications such as metal material processing, sugar processing, wood piece processing, quarry, geology, mining and mineral processing, which requires that particles in images have to be delineated online. This paper shows that a method, involving image evaluation and edge based particle delineation, is a highly efficient way of delineating particles online. No earlier work on delineating particles online has exploited these two building blocks for making robust delineation. Our method has been tested experimentally for different kinds of particle images those are difficult to detect by ordinary edge detection algorithms. The reason for the powerfulness of the technique is that image evaluation and particle delineation are highly cooperative processes. As tested, the algorithms can be applied into many similar engineering areas.

1 Introduction

In many online particle image-processing applications, the common problem for the particle image processing is to delineate every particle, but in most industrial cases, particle images are difficult to segment due to rough surface, overlapping, and size variation etc. If considering the aspects of the algorithms or methods of image segmentation, the existing systems could be classified into at least four classes: (1) thresholding on histogram of gray levels; (2) boundary tracing or edge detection; (3) region growing or merge & split; and (4) thresholding and granulometry (= morphology segmentation on a binary image, e.g. Watershed segmentation algorithm). Since a large variation of particle images, there is no segmentation algorithm that can process all kinds of particle images. In brief, edge detection tends to produce too few or too many boundary candidates, intensity similarity does not fit the data since grey value variation between particles is small and within-particle variation quite large, and watershed algorithms based on using maxima within particles do not perform as well as the approach in this paper.

For online detection of rock particles, Gallagher[1] in 1976 developed the earliest image analysis system. In his PhD study, he set up a system aimed to

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 468–477, 2008.

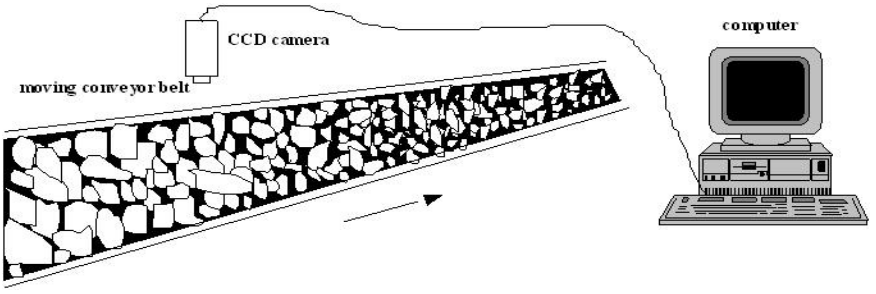


Fig. 1. The online system configuration

measure particle size parameters on a conveyor belt. The figure 1 shows the system configuration. The camera was mounted above the particle stream with its optical axis aligned normal to the moving bed of particles. The size distribution of the particles was then computed by finding the spacing of edges with a chord sizing method. In most applications, the quality of particle images varies too much, which make image segmentation hard. Therefore, this research subject becomes a hot topic in the world during last thirty years. Today, a number of image systems have been developed for measuring particles in different application environments such as particles on/in gravitational flows, conveyor belts, rockpiles, and laboratories. The research and development has been and is being carried out in many countries, the detailed information can be found in [2] [3] [4] [5].

Rock particle images taken from a fast moving conveyor belt vary so much that the quality of any two successive image frames are not the same. For example, a) an image may include about 80% fine materials which is difficult to recognize by a segmentation program; b) an image may consist of only a few particles which is of less interest to analyze; c) some images are very dark or very bright with poor contrast of gray values, in which case erroneous image information will be obtained by a segmentation program, and the result of image analysis will be affected seriously; d) some images are quite blurry, caused by an increase of the speed of conveyor belt or other reasons, which can also be difficult to process, etc. Hence, classification with respect to a number of image categories, intimately linked to the feasibility of successful segmentation, is very important for this application. A kind of inspection system needs to be set up. Inspection tasks should be performed in real-time, and complex and time-consuming texture analysis cannot be used. Thus, we avoid doing complex texture analysis.

Empirically, from a performed field investigation, there seems to be at least four defect classes of rock particle images from a moving conveyor belt (By "defect", we mean "not proper or particularly suitable to process further"), namely: (1) empty conveyor belt images (or few particles on a belt); (2) large particles images (a few large particles are included in a image, but they overlap each other or part of them are not included in the image); (3) blurry images (most of the edges are lost, or weak); (4) fine material images (on average, each particle only occupies a few image pixels). Some of these defect classes are difficult to

recognize even by human vision. If a segmentation algorithm accepts them, we should not expect segmentation results to be satisfactory.

The approach to automatically, simply and quickly recognize and distinguish these four classes can be based on global statistical information. For this purpose, we have used histograms of both the original image and its gradient magnitude image as input to a rock particle classification procedure that makes the system automatically select non-defect rock particle image for further segmentation or average size estimation. The general program sequence is: (1) image pre-processing for smoothing original image and creating gradient magnitude image; (2) calculation of mean gray values of smoothed rock particle image and corresponding gradient magnitude image, the standard deviation and skewness of the gradient magnitude image; and (3) rock particle image classification based on statistical texture analysis of both the original image and its gradient magnitude image. This procedure has been tested both in laboratory and in the field, and seems rather promising.

2 Particle Image Evaluation

One approach to rock particle image classification is a scheme intended to work for the case of washed rock particle material transported on a dark conveyor belt, assuming that the rock particle color is brighter than the wet conveyor belt. The classification program is also developed to serve as a kind of pre-process for subsequent image segmentation for increasing its accuracy. Hence, when the system grabs one image frame, the system should judge if the image should be processed. If the image quality is poor, it belongs to the defect classes. It is not possible or desirable to conduct analysis by a segmentation algorithm, and therefore the image should be omitted and the segmentation algorithm should wait for the next image frame.

For industrial applications, during the working period of conveyor belt, one also wants to know: (1) the size distribution of rock particle particles, which includes the percentages of smallest particles and largest particles; (2) the situation of rock particle feedback for the crusher, the relative information can be obtained from the percentage of the number of images of an empty conveyor belt; (3) the variation of conveyor belt speed, which can be estimated crudely by monitoring the percentage of blurred images.

Based on the field investigation and the above motivations, we have classified rock particle images into five basic classes, four of them being the images unacceptable for the segmentation algorithm (defect classes). They are images of empty conveyor belt, fine materials, large particles and images affected strongly by motion blur. The remaining ones are acceptable images for the segmentation algorithm. For the defect classes, the basic considerations are described as follows.

We use the following notation: μo is the mean value of the smoothed gray level image, and μm= mean value of the corresponding gradient magnitude image. σo, σm are the standard deviations of the original and gradient magnitude

images, respectively. The various kinds of threshold values are $\lambda 0$, $\lambda 1$, $\lambda 2$, etc. For example, the threshold $\lambda 0$ is linked to the inequality $\mu o \geq \lambda 0$.

For an empty conveyor belt, μo will be substantially lower compared to the case when the belt is almost filled with rock particles, provided the belt is noticeably darker than the rock particles. This would tend to be the case if the belt is washed, because washed conveyor belts are much darker than dry ones. Dry belts are often sprinkled with dust making them both brighter and full of traces and patches of dust and dirt. It is advisable to have washed conveyor belt, since this makes both background-particle segmentation and particle delineation in subsequent particle segmentation easier. In what follows, we will assume that the belt is darker than rock particles. By "empty" conveyor belt, we mean that not more than 20% of the image area is occupied by rock particles (The case when the belt is filled between 20% and 80% we have not dealt with yet, except some preliminary experiments), and if the belt is "empty", there will be very few edges implying relatively lower average gradient magnitudes, i.e., μm is relatively lower. For images strongly affected by motion blur, edges are weak, making μm low. But μo need not be low (blurry images of almost empty conveyor belt wind up in the "empty-class"). To distinguish between the cases of quite small particles, medium-sized particles and large particles, edge density seems to be an efficient tool, provided rock particles occupy at least 80% of conveyor belt, see Table 1, and the column of $qm/\sigma m$.

Based on these considerations, we have chosen eight typical rock particle images for statistical analysis (see Fig. 2 and Table 1). The rock particle materials come from the quarry of Underås in Södertälje, south of Stockholm.

In experiments presented in Table 1, Fig. 1, Fig. 2, we noted that although both μm and μo are relatively lower for "empty" belt, and μm lower for strongly blurry images, it is somewhat hard to choose the threshold independently of material (illumination was more or less fixed). Some rock particle materials are darker, some brighter. On the other hand, it is clear that some kind of normalized average, say $\mu m/\mu o$ is almost independent of material, in our experiments. It makes sense to normalize μm by μo, since we then obtain a quantity that is something like average edge strength divided by (normalized by) mean gray value in an image.

From Fig. 2 and Table 1, the statistical result from laboratory tests is analyzed as follows:

Class 1 (the images with empty conveyor belt, as shown in the image No. 1): the values of μo and $\mu m/\mu o$ are comparatively lower.

Class 2 (the images with blur, as shown in the images No. 2-3): the ratio of $\mu m/\mu o$ is lower than the other values in the column, but μo is not.

Classes $3 \sim 5$ (the rock particle images, as shown in the images No. 4-8): the values of $qm/\sigma m$ which is related to rock particle edge density is lower, when the average size of particles is smaller. Other values such as $\mu o/\sigma o$, $\mu m/\mu o$ and $\mu m/\sigma m$ also seem to decrease with increasing average size of rock particle particles. Table 2 shows experimental, empirical classification criteria for classifying the above images of Fig. 2.

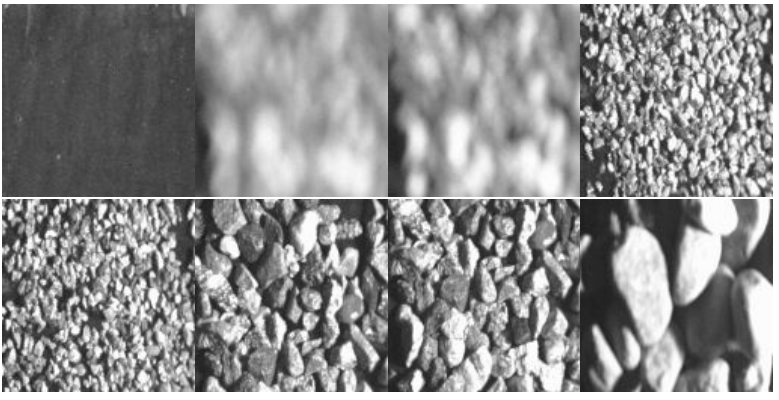


Fig. 2. Different classes of rock particle images (see Table 1)

Table 1. The parameters statistics of the eight typical rock particle images in Fig. 2

#.	μo	σo	μo /σo	μm	μm /μo	σm	μm /σm	qm	Qm /σm	sieving size mm	class
1	77.37	6.58	11.76	15.28	0.20	16.50	0.93	26.23	1.59		1
2	171.47	36.36	4.72	33.77	0.20	28.83	1.17	41.01	1.42	8 $\sim$ 12	2
3	166.00	43.17	3.85	52.62	0.32	36.21	1.45	41.53	1.15	8 $\sim$ 12	2
4	146.25	50.02	2.92	161.49	1.10	75.58	2.14	-47.63	-0.63	2 $\sim$ 4	3
5	142.76	48.67	2.93	160.23	1.12	75.15	2.13	-45.35	-0.60	2 $\sim$ 4	3
6	131.78	57.85	2.28	134.73	1.02	81.82	1.65	44.45	0.54	8 $\sim$ 12	5
7	139.41	61.09	2.28	136.27	0.98	82.29	1.66	41.86	0.51	8 $\sim$ 12	5
8	135.33	80.77	1.68	79.39	0.59	85.44	0.93	85.61	1.00	32 $\sim$ 64	4

Table 2. The classification criteria of defect images for the images in Fig. 2

class	μo	μm/μo	qm/σm
1 (empty)	$<$ λ0 (here λ0= 85)	$\leq$ λ1 (here λ1= 0.4)	
2 (blur)	$>$ λ2 (here λ2= 105)	$\leq$ λ1	
3 (fine)			$<$ λ4 (here λ4= 0.1)
4 (large)			$>$ λ3 (here λ3= 0.8)

Summing up, this approach is developed based on a statistical texture analysis of both original rock particle image and gradient magnitude image. It is simple and works fast, but is crude. The parameters λ0 $\sim$ λ4 are not necessarily constant values for any kind of applications of rock particle materials. Experience and rock particle material types determine them, because different materials have different colors and surface roughness etc. The following description presents an

example of how to use this approach to automatically or semi-automatically select acceptable rock particle images from a moving conveyor belt.

3 Particle Image Segmentation Algorithm

After image evaluation, the selected images (fine, large) will e processed. Figure 3 shows (a) a grey value landscape over layered with a sample point grid. A simple edge detector uses differences in two directions:$\Delta_x = g(x+1, y) - g(x, y)$, $\Delta_y = g(x, y+1) - g(x, y)$. In the valley detector, we use four directions. Obviously, in many situations, the horizontal and vertical grey value differences, do not characterize a point, such as P, well. See Fig. 3.

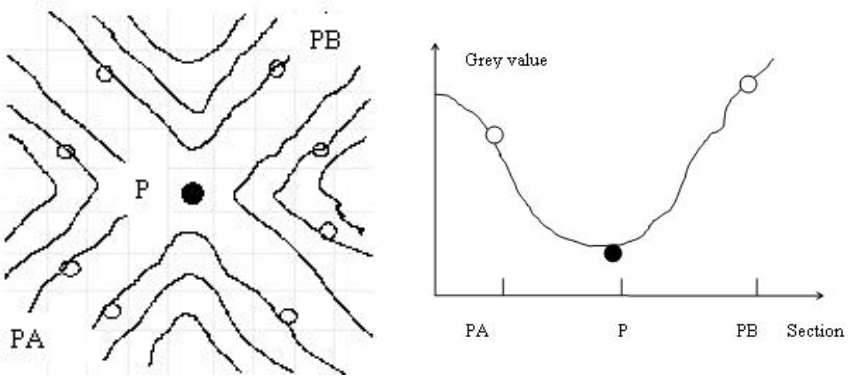


Fig. 3. Examine the point P, determining if it is a valley pixel, or not. Circles in the sparse (i, j)-grid. It moves for each P ∈ (x, y)-grid. (a) A grey value landscape over layered with a sample point grid. (b) PA-PB section.

In the example of Fig. 3, we see that P is surrounded by strong negative and positive differences in the diagonal directions: $\nabla_{45} < 0$, and$\Delta_{45} > 0$,$\nabla_{135} < 0$, and $\Delta_{135} > 0$, whereas, $\nabla_0 \approx 0$, and $\Delta_0 \geq 0$,$\nabla_{90} \approx 0$, and $\Delta_{90} \approx 0$. Where $\Delta$are forward differences: $\Delta_{45} = f(i+1, j+1) - f(i, j)$, and ∇ are backward differences: $\nabla_{45} = f(i, j) - f(i-1, j-1)$, etc. for other directions. We use $\max(\Delta_\alpha - \nabla_\alpha)$ as a measure of the strength of a valley point candidate. It should be noted that we use sampled grid coordinates, which are much more sparse than the pixel grid$0 \leq x \leq n$, $0 \leq y \leq m$. f is the original grey value image after weak smoothing. What should be stressed about the valley edge detector is:

(a) It uses four instead of two directions;

(b) It studies value differences of well-separated points: the sparse $i \pm 1$ corresponds to $x \pm L$ and $j \pm 1$ corresponds to$y \pm L$, where$L >> 1$, in our case, $3 \leq L \leq 7$. In applications, if there are closely packed particles of area ¿ 400 pixels, images should be shrunk to be suitable for this choice of L. Section 3 deals with average size estimation, which can guide choice of L;

(c) It is nonlinear: only the most valley-like directional response $(\Delta_\alpha - \nabla_\alpha)$ is used. By valley-like, we mean $(\Delta_\alpha - \nabla_\alpha)$ value. To manage valley detection in cases of broader valleys, there is a slight modification whereby weighted averages of $(\Delta_\alpha - \nabla_\alpha)$- expressions are used.$w_1\Delta_\alpha(P_B) + w_2\Delta_\alpha(P_A) - w_2\nabla_\alpha(P_B) - w_1\nabla_\alpha(P_A)$, where,$P_A$ and P_B are shown in Fig. 2. For example, $w_1 = 2$ and $w_2 = 3$ are in our experiments.

(d) It is one-pass edge detection algorithm; the detected image is a binary image, no need for further thresholding.

(e) Since each edge point is detected through four different directions, hence in the local part, edge width is one pixel wide (if average particle area is greater than 200 pixels, a thinning operation follows boundary detection operation);

(f) It is not sensitive to illumination variations, as shown in Fig. 5, an egg sequence image. On the image, illumination (or egg color) varies from place to place, for which, some traditional edge detectors (Sobel and Canny etc.) are sensitive, but the new edge detector can give a stable and clear edge detection result comparable to manual drawing result.

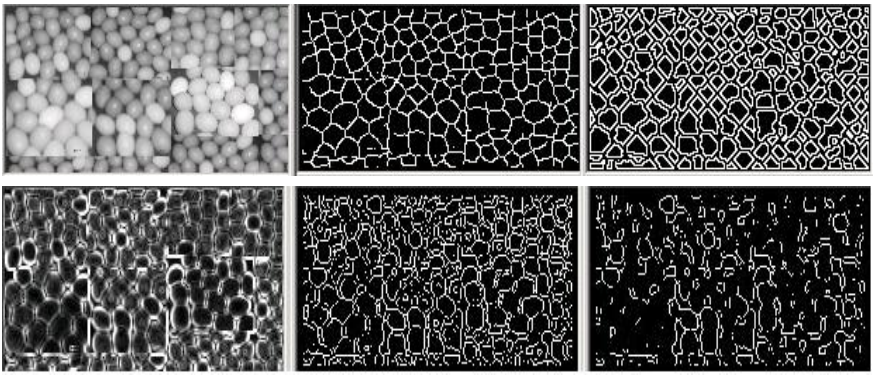


Fig. 4. Egg image test: (a) original image (400x200 pixels), (b) new algorithm result, (c) manual drawing result (180 eggs), (d) Sobel edge detection result, and (e) and (f) Canny edge detection results with different thresholds

After valley edge point detection, we have pieces of valley edges, and a valley edge tracing subroutine, filling gaps is needed (Some thinning is also needed.).

As a background process, there is a simple grey value thresholding sub-routine, which before classification creates a binary image with quite dark regions as the bellow-threshold class. If this dark space covers more than a certain percentage of the image, and has few holes, background is separated from particles by a Canny edge detector [11] along the between-class boundaries.

To test the delineation algorithm, we have taken a number of different particle images from a laboratory, a muckpile, and a moving conveyor belt. In this section, we just present three different particle images to show representative segmentation results.

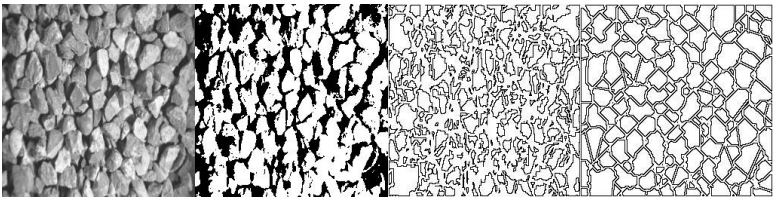


Fig. 5. A densely packed particle image and segmentation results: (a) Original image (512x512 resolution, in a lab, Sweden), (b) Auto-thresholding, (c) Segmentation on similarity, and (d) New algorithm result

As an illustration of typical difficulties encountered in particle images we show an example, where thresholding [7] and similarity-based segmentation [9] are applied. Figure 5 shows one original image of closely packed particles. in which most particles are of medium size, according to the classification algorithm. Under-segmentation and over-segmentation normally take place in Fig. 5(b) and Fig. 5(c). The new algorithm shows a better result in Fig. 5(d).

Experimental image results with the new algorithms are shown in Figs. 6-8 too, where particle images are classified into classes of small particles, and using the developed segmentation algorithm segments medium sizes with non-void spaces and the images. In Figs 6-7, the size of an image is 512x512 pixels, and the numbers of particles in an image are 1560 and 276. The segmentation results are quite good, and comparable to human performance.

Fig. 6. A densely packed particle image from a rockpile (USA). Image resolution is 512x512.

In addition to particle images, the developed algorithm can also be used for other application images. Fig. 8 shows the testing result for froth images in mineral processing. As tested and compared, the processing speed of the algorithm is hundred times faster than Watershed algorithm, it is suitable for online system too.

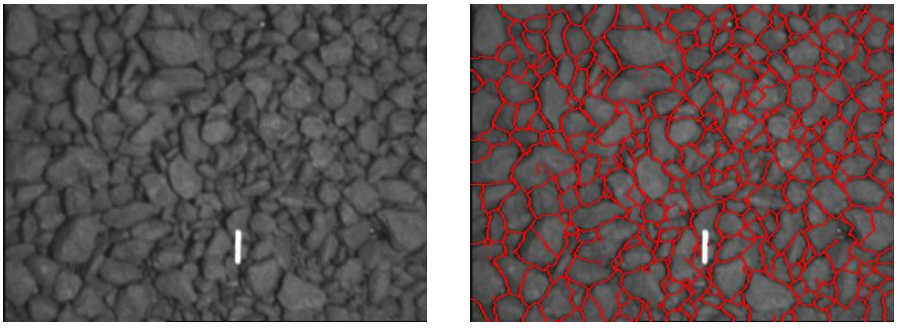


Fig. 7. A densely packed coal particle image from a underground mine (Africa). Image resolution is 512x512.

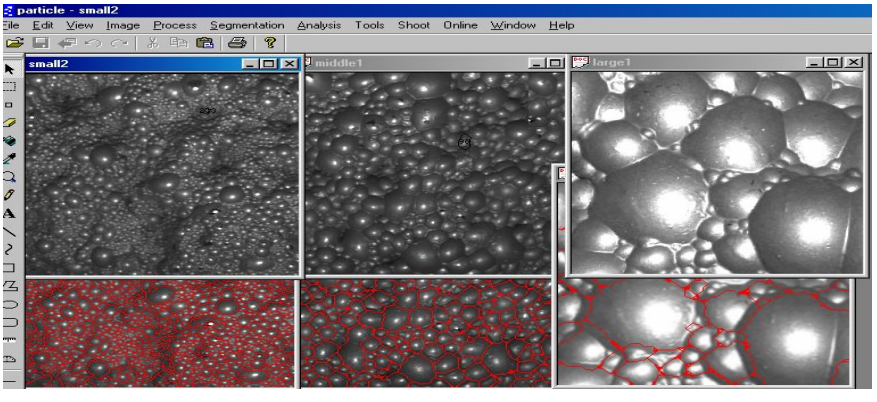


Fig. 8. Froth images (Sweden). Image resolution is 256x256. Three images are classified into Classes 1-3.

4 Conclusion

In this paper, a new type of online particle delineation method has been studied and tested; the combination of image evaluation algorithm and particle delineation algorithm has been described. The particle image evaluation algorithm was developed based on a statistical texture analysis approach using both original and gradient magnitude images. The particle delineation algorithm studied is actually based on both valley-edge detection and valley-edge tracing. The presented particle delineation algorithm seems robust for densely packed complicated objects (e.g. rock particles on a moving conveyor belt). The method can be used into other applications such as froth images in mineral processing.

References

1. Gallagher, E.: Optoelectronic coarse particle size analysis for industrial measurement and control, Ph.D. thesis, University of Queensland, Dept. of Mining and Metallurgical Engineering (1976)
2. Ord, A.: Real-time image analysis of size and shape distributions of rock fragments. In: The AusIMM, Explosives in Mining Workshop, Melbourne, Australia, pp. 115–119 (1988)
3. Lin, C.L., Yen, Y.K., Miller, J.D.: Evaluation of a PC Image - Based On - Line Coarse Particle Size Analyzer. In: Proceedings of Emerging Computer Techniques for the Mineral Industry Symposium, AIME/SME, pp. 201–210 (1993)
4. Kemeny, J., Mofya, E., Kaunda, R., Lever, P.: Improvements in Blast Fragmentation Models Using Digital Image Processing, Fragblast. 6(3-4), 311–320 (2002)
5. Maerz, N.H., Palangio, T.W.: Post-Muckpile, Pre-Primary Crusher, Automated Optical Blast Fragmentation Sizing, Fragblast. 8(2), 119–136 (2004)
6. Pal, N.R., Pal, S.K.: A review of image segmentation techniques. J. Pattern Recognition 26(9), 1277–1294 (1993)
7. Wang, W.X.: Image analysis of aggregates. J. Computers & Geosciences 25, 71–81 (1999)
8. Stephansson, O., Wang, W.X., Dahlhielm, S.: Automatic image processing of fragments. In: ISRM Symposium: EUROCK 1992, Chester, UK, 14 -17 September, pp. 31–35 (1992)
9. Otsu, N.: A threshold selection method from gray-level histogram. IEEE Trans. Systems Man Cybernet SMC-9, 62–66 (1979)
10. Chung, S.M.: A new image segmentation technique based on partition mode test. J. Pattern Recognition 16(5), 469–480 (1983)
11. Canny, J.F.: A computational approach to edge detection. J. PAMI-8, 6 (1986)
12. Spiegel, M.R.: Schaum's outline series, Mathematical Handbook of Formulas and Tables, 28th printing, U.S.A (1992)

Improving Inter-cluster Broadcasting in Ad Hoc Networks by Delayed Flooding

Adrian Andronache, Patricia Ruiz, and Steffen Rothkugel

FSTC, Campus Kirchberg, 6, rue Richard Coudenhove-Kalergi,
L-1359 Luxembourg, Luxembourg
{adrian.andronache,patricia.ruiz,steffen.rothkugel}@uni.lu

Abstract. Clustering is a well-known approach to cope with mobility in multi-hop ad-hoc networks. The aim of clustering hereby is to detect and maintain stable topologies within a set of mobile nodes. WCPD is an algorithm that allows to interconnect multiple stable clusters and to exchange data across single cluster boundaries. However, disseminating information beyond stable connected sets of nodes is required in some settings as well. The DWCF algorithm introduced in this paper aims at high coverage while keeping the overhead low. DWCF facilitates that by exploiting the cluster structures and selectively forwarding data to foreign neighbor clusters.

Keywords: Networking, ad hoc, mobile, delay, tolerant, weighted, cluster, broadcast.

1 Introduction

An increasing number of present-day mobile devices like tablets, PDAs, smartphones and laptops provide adapters for both, ad hoc and backbone communication. In our hybrid network model mobile devices use Wi-Fi adapters to communicate in an ad hoc fashion and cellular adapters to access an infrastructured backbone. The backbone assists the mobile network by providing authentication and partition inter-connection mechanisms as well as information of interest. In both ad-hoc networks and sensor networks, clustering is one of the most popular techniques for locality-preserving network organization [1]. Cluster-based architectures effectively reduce energy consumption, and enable efficient realization of routing protocols [2], data aggregation [3,4], and security mechanisms [5]. In our previous introduced backbone-assisted mobile ad hoc network applications [6,7] a cluster is created by electing a so called clusterhead among one hop neighbor devices. The applications use the clusterheads to keep track of cluster interests and register them at the backbone by an uplink maintained for instance by a cellular connection. The backbone provides multimedia items of interest to the registered clusterheads which forward them to interested cluster slave devices in the ad hoc network. For the self-organization of the mobile devices in cluster structures we employ the Node and Link Weighted Clustering Algorithm – NLWCA [8] . NLWCA is designed to organize mobile ad

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 478–487, 2008.

hoc networks in one hop clusters. Each device elects exactly one device as its clusterhead, i.e. the neighbor with the highest weight. The weight of the own device is calculated by NLWCA based on device properties that are important for the application running on top of it. For instance good signal strength to the backbone network is important for a multimedia providing application since it increases the download bandwidth. Favorable is also a high network degree, which optimizes the multimedia items spreading process. These device properties along with the battery power are used to calculate the weight, which is sent by the beacon to the one-hop neighbors. An elected clusterhead also investigates its one-hop neighborhood, similarly electing the device with the highest weight as its clusterhead. This process terminates when a device elects itself as its own clusterhead, due to the fact of having the highest weight among all its neighbors. We call all intermediary devices along such clusterhead chains sub-heads. Each device on top of a chain is called a full clusterhead, or, in short, clusterhead. Hence, in each network partition, multiple clusterheads might coexist. The main goal of the algorithm is to avoid superfluous re-organization of the clusters, particularly when clusters cross each other. To achieve this, NLWCA assigns weights to the links between the own node and the network neighbor nodes. The link weight is used to keep track of the connection stability to the one-hop network neighbors. When a link weight reaches a given stability threshold the link is considered stable and the device is called stable neighbor device. The clusterhead is elected only from the set of stable neighbors which avoids the re-organization of the topology when two clusters are crossing for a short period of time. In previous work we introduced the Weighted Cluster-based Path Discovery protocol (WCPD) [9] , which is designed to take advantage of the cluster topology built by NLWCA in order to provide reliable path discovery and broadcast mechanisms in mobile ad hoc networks. In this work we present an approach that increases the ad hoc network inter-cluster broadcast performance of the WCPD protocol by using delayed flooding. The remaining of the paper is organized as follows: Section 2 describes the WCPD inter-cluster broadcasting mechanism. Section 3 introduces the *Delayed Weighted Cluster Flooding (DWCF)* algorithm. In Section 4 we describe the simulation settings and the results obtained. Related work is located in Section 5. Section 6 presents the conclusions and the future work.

2 WCPD Inter-cluster Broadcasting

In this section we describe the WCPD broadcasting protocol introduced in [8]. Since DWCF aims to improve the broadcast performance in mobile ad hoc networks, we present the WCPD protocol that works without the assistance of the backbone. WCPD runs on each network node and requires solely information available locally in the one hop neighborhood. The algorithm uses information provided by NLWCA: the set of stable connected network neighbor nodes and the ID of the own clusterhead. NLWCA also propagates by beacon the own weight

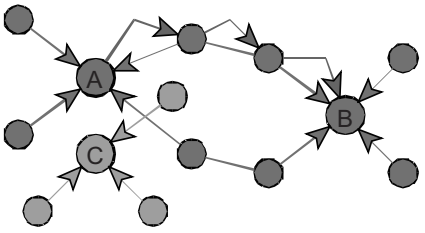


Fig. 1. The clusterheads A and B are stable connected in the topology built by NL-WCA. The main goal of the WCPD broadcasting protocol is to reach the nodes of stable connected clusters. To achieve this, when A broadcasts a message it also sends it by multi-hop unicast (illustrated by the arrows) to the stable connected cluster B. As side effect, the crossing clusterhead C receives the broadcast message from A like every other node in one-hop communication range. Clusterhead C re-broadcasts the message in order to reach the stable connected nodes, thus increasing the number of the broadcast receivers..

and the ID of the current clusterhead. Besides the information provided by NL-WCA, the WCPD protocol uses the beacon to disseminate the list of locally discovered nearby connected clusterheads. By doing so, every node has the following information about each stable one hop neighbor: its clusterhead ID and the ID set of discovered clusterheads and the respective path length. After the data of all stable one hop neighbors is checked, the set of discovered nearby clusterheads and the path length is inserted into the beacon in order to propagate it to the one hop neighborhood. Since WACA elects one clusterhead in each one-hop network neighborhood, the path length between two clusterheads of connected clusters is at least two hops and at most three hops. If the number of hops to a clusterhead is higher than three then its cluster is not directly connected to the local cluster and its ID is not added to the set of nearby clusterheads. In other words, WCPD keeps track only of stable connected clusterheads that are at most three hops afar. The WCPD broadcasting algorithm is simple and easy to deploy: the broadcast source node sends the message to the clusterhead, which stores the ID of the message and broadcasts it to the one hop neighborhood. After that, it sends it to all nearby clusterheads by multi-hop unicast [1]. The inter-cluster destination-clusterheads repeat the procedure except that the message source clusters are omitted from further forwarding. Additionally the information about the ID of the broadcast messages and their sources is stored for a given period of time to avoid superfluous re-sending of the message. The protocol sends the broadcast message to nearby clusters connected by stable links in order to disseminate it to the network partition. Nevertheless the message also reaches crossing clusters since the broadcasts are received by all nodes in the one-hop neighborhood of local leaders. This increases the chance that the message reaches a high number of nodes in the mobile network partition.

3 The Delayed Weighted Cluster Flooding Protocol (DWCF)

The inter-cluster broadcast protocol of WCPD is designed to reach the members of stable-connected clusters in the network vicinity of the broadcast source node. The algorithm is easy to deploy and aims to avoid computational and communication load on the mobile nodes. To achieve this, the next hop to a cluster in the vicinity is picked up based on its weight, thus the device with the highest amount of energy left is elected as router. This method is simple but it does not take topological properties of the potential next-hop nodes into account. Since the protocol acts in mobile environments the elected router-node might lose its path to the destination cluster, thus dropping the message. Further, the mechanism is not well suited for applications that require network flooding since the main goal of the WCPD inter-cluster broadcast is to reach only the nearby stable-connected clusters. In order to improve the network flooding performance of the WCPD protocol we introduce the *Delayed Weighted Cluster Flooding (DWCF)* algorithm, which is inspired by the Delayed Flooding with Cumulative Neighborhood (DFCN) algorithm introduced in [10]. The DFCN algorithm aims to minimize the network load during flooding by taking into account the network density. When the network is sparse, it is quite difficult to spread a message, so in that conditions, a node should forward the message as soon as another device enters the communication range. This would lead to good results since every re-emission proves to be useful because of the reduced number of meeting points between nodes. However, in dense networks this strategy would lead to a high overload and broadcast storms. In order to avoid this, DFCN sets a random delay (RAD) when a node receives a new message. If the density is low then the RAD is immediately set to zero after a new neighbor is met, but this behavior is disable when the density is high. This perception of the density corresponds directly to its neighborhood and it is managed with a threshold called densityThreshold. DFCN attaches to the broadcast message a list T(m) containing the current neighbors of the node. This list is managed as follows: when a node s broadcasts a message m to its neighbors, it assumes that all of them will receive m. Therefore, T(m) is set with all the neighbors, N(s), of the sender-node s. Once the RAD is finished, DFCN uses this list to decide whether the received message will be forwarded or not, in terms of the neighbors that already received the message which are in the one hop neighborhood. For that, a threshold called minBenefit is set, which is formally defined on the benefit, computed as the ratio between the neighbors of s which do not belong to T(m), and the total neighbors of s, N(s). The higher the benefit, the higher the probability of (re)-emission of a message. The Delayed Weighted Cluster Flooding algorithm aims to be as simple as possible but at the same time to improve the flooding performance of the WCPD protocol. To achieve this, the protocol uses the cluster topology provided by NLWCA and the cluster information provided by the WCPD Nearby-Cluster Discovery protocol. The basic idea of the protocol is to use broadcast instead of unicasts to reach nearby clusters. This increases the chance that a sent message reaches the destination cluster even if the connecting cluster-border nodes are

changing during transmission. Also, the broadcast might reach nodes belonging to different clusters, thus reducing the communication load. One of the main design goals is to keep the algorithm as simple as possible, which reduces computational load on the nodes, but at the same time to avoid communication overload. The main issue is to avoid that several nodes situated on the border of the cluster re-broadcast a message in order to reach the same neighbor clusters. Inspired by DFCN, the DWCF algorithm uses random delays before it forwards the broadcast messages. Further, the decision to re-send the message is based solely on the presence of foreign-cluster nodes in the one-hop neighborhood at the end of the delay. This keeps the algorithm simple and reduces the number of node IDs added to the broadcast message since generally the number of nearby clusterheads is much lower than the number of one-hop neighbors. The broadcast-handler method is called when a node receives a broadcast message. First of all DWCF checks if the message was already processed (delay = -1) and breaks the method if true. If the message was not processed but is already in the queue then the set of receiver cluster IDs is updated since the sender might reach new clusters. This avoids later that nodes re-send the message if some other node already sent it during the delay to all foreign clusters in one-hop neighborhood. In case the message is new, it is instantly re-broadcasted if the receiver is a clusterhead or a sub-head in order to reach all nodes in the cluster. If the receiver is a slave node then a random delay is set for the message, which is added to the queue and the delay timer is started. The delay timer fires a tick event every second. The tick handler method is in charge of decreasing the delay of every message in the queue that is not already processed (having a delay of -1). If the delay of a message reaches the value 0 then the DelayBroadcast method is called. If the queue contains no message with delay higher than 0 then the timer is stopped in order to save computational load. After the delay of a message, the DelayBroadcast method is called. DWCF searches the last received beacons of the one-hop neighbor nodes in order to discover foreign clusters. This information is provided by NLWCA, which uses the beacon to send the own weight node and the current clusterhead ID to the neighboring nodes. The set of discovered foreign cluster IDs is compared to the set of receiver-cluster IDs contained in the message. If at least one new cluster is reachable then DWCF replace the receiver-cluster IDs with the current neighboring cluster IDs and broadcasts the message. The message is also broadcasted if it was not already received from a member node of the own cluster. Thus, the messages received from foreign clusters reach the own clusterhead for further spreading. After the message is forwarded its payload is deleted but the ID is kept for a given time to avoid further processing. The pseudo code of the algorithm is like follows:

```
--- Required Data ---
d: The ID of the node.
C(d): The clusterhead ID of d.
N(d): The set of one-hop neighbor nodes of d.
m(d): A broadcast message received from d.
M(m): The ID set of cluster reached by broadcast of m.
```

```
f(m): True if m was received from the own cluster.
r(m): The re-send delay for m.
Q: The message queue.
T: Delay-timer, fires a tick event every second.

--- Handler method m(e) on node d ---
If( r(m) == -1 ) return; // already processed message
If( Q contains m ) do: // message already in queue
   Merge received M(m) with local M(m);
   return; // break
End do;
If( d is clusterhead or is sub-head ) do:
   r(m) = -1; // mark it as processed
   add (m,r(m)) to Q; // remember it
   Send_broadcast(m); // instantly re-send
   return;
End do;
If( C(e) equals C(d) ) f(m) = true;
r(m) = random number between 1 and n;
add (m,r(m)) to Q;
If( T is not running ) start T;

--- Handler method for the ticks of T ---
For each ( m(e) in Q ) do:
   run = false; // timer re-start flag
   If( r(m) > -1 ) r(m) = r(m) - 1;
   If( r(m) > 0 ) run = true;
   If( r(m) == 0 ) DelayBroadcast(m);
End do;
If( run = false ) stop T; // more messages to process

--- DelayBroadcast method for message m on node d ---
For each ( e in N(d) ) // find foreign clusters
   If( C(e) not equals C(d) ) add C(e) to K;
re-send = false;
If( K \ M(m) not null ) re-send = true;
If( f(m) == false ) re-send = true;
If( re-send == true ) do:
   M(m) = K;
   Send_broadcast(m);
   Delete payload(m); // keep only the ID
End do;
```

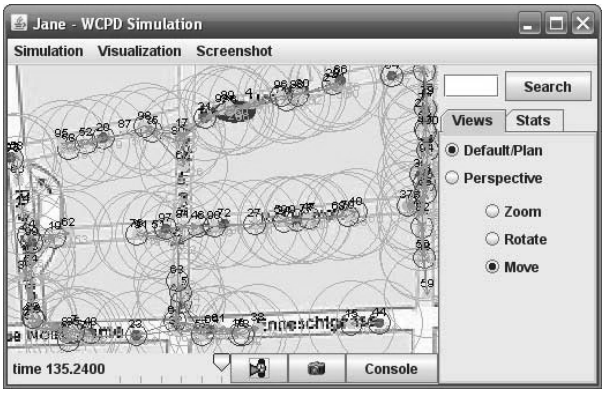


Fig. 2. JANE simulating the WCPD protocol on 100 devices. The mobile devices move on the streets of the Luxembourg City map. The devices move with a speed of 0.5 – 1.5 m/s and have a sending radius of 20 m.

4 Experiments and Results

In order to compare the flooding performance of the two algorithms, we implemented them on the top of the JANE simulator [11] and performed several experiments. For the conducted experiments we used the Restricted Random Way Point mobility model [12], whereby the devices move along defined streets on the map of Luxembourg City 2. For each device the speed was randomly varied between 0.5 and 1.5 m/s (1.8 and 5.4 km/h), which are common human speeds. For this speed range the NLWCA link stability threshold is set on 2. At simulation startup, the devices are positioned at random selected crossroads and the movement to other crossroads is determined by the given random distribution seed. A number of ten different random distribution seeds were used in order to feature results from different topologies and movement setups. In order to monitor the information dissemination performance and network load of the broadcasting mechanisms, a node was chosen to broadcast a message every 10 seconds during different simulation runs using different distribution seeds. The number of sent messages (i.e. broadcasts and unicasts) during the dissemination and the number of reached network nodes were tracked for 300 seconds. The experiments were done in sparse networks with 100 mobile devices and in dense networks with 300 devices.

The tracking results regarding the message dissemination performance and network load of the inter-cluster broadcasting protocols are presented in Fig. 3 and Fig. 4. The overall results show that DWCF performs much better in terms of message dissemination than the WCPD inter-cluster broadcast protocol. The denser the network, the higher the difference between both the number of sent messages and the number of receiver nodes. The results for sparse networks with 100 devices are presented in Fig. 3. The WCPD uses a number of messages with a mean value around 100 to reach around 10% of the receivers. The low

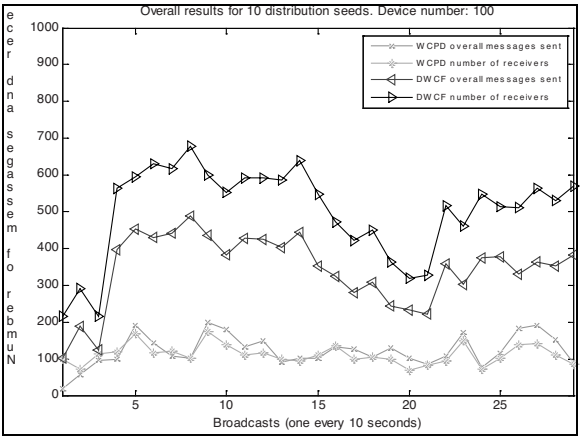


Fig. 3. Simulation results for sparse networks with 100 devices. The number of sent messages is almost the same as the number of received when using WCPD inter-cluster broadcasting. The flooding performance of DWCF is much higher but at the cost of a higher network overload.

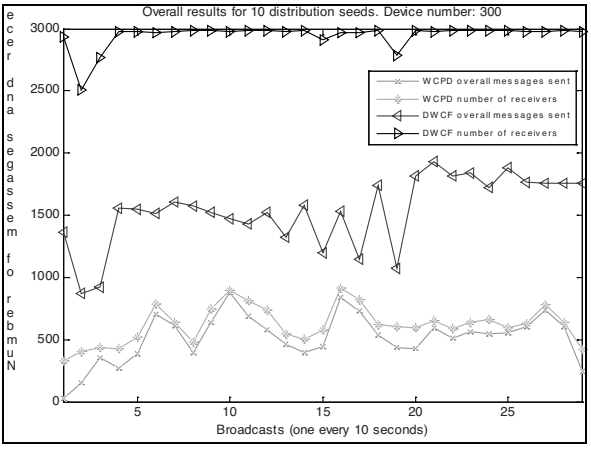


Fig. 4. In dense networks with 300 devices the performance of DWCF is even better. The difference between the number of sent messages and the number of receivers is much higher than in sparse networks.

reachability can be explained by the fact that WCPD aims to reach with the broadcast only the stable connected clusters. The number of sent messages is high in relation to the number of reached devices and it emerges due to the multi-hop unicasts used to forward the message to connected clusters. The DWCF protocol performs much better in terms of reached devices whose ratio has a mean value around 50%. The increased dissemination performance requires a higher number of sent messages, which has a mean value around 400.

The results in Fig. 4 show that the DWCF performs even better in denser networks with 300 devices, where only 1500 messages were sent. The most of them reached 100% of the receivers. Since DWCF uses broadcasts to forward the messages to neighbor clusters, the probability to reach them is higher than when using multi-hop unicasts. Even if the connecting nodes change during the forward process, the new ones receive and re-send the broadcasted message. The broadcasted messages also reach crossing foreign clusters in communication range of the forwarding border nodes. These clusters are missed by the WCPD unicasts.

5 Related Work

The Delayed Flooding with Cumulative Neighborhood (DFCN) [10] is a broadcasting protocol designed for mobile ad hoc networks. The protocol uses the information about the one hop neighbors provided by the ad hoc network beacons. When a node receives a broadcast message a random delay is set for the re-sending of the message. The protocol aims to improve the flooding performance by forwarding the message instantly when a new node is detected in sparse networks and to wait longer in dense networks in order to avoid communication overload. DFCN is optimized in [13] using cMOGA, a new cellular multi-objective genetic algorithm. Main goals of the work are minimizing the duration of the broadcasting process, maximizing network coverage, and minimizing the network usage. Three different realistic scenarios were used corresponding to a shopping center, the streets in a city and a wide non-metropolitan area wherein several roads exist. In [14] a delay-tolerant broadcasting algorithm is proposed for wide and public wireless dissemination of data. The approach is not using flooding or routing, it uses only the mobility of the nodes in order to spread the data. The hybrid setting of Sun et al [15] consists of base stations that are inter-connected and mobile devices that can connect locally via an ad-hoc mode or to a base station if near enough to it. Two routing schemas are introduced to deal with different application requirements. Sun et al research points out that the efficiency of the chosen communication mode strongly depends on the applications running in the overall network.

6 Conclusion and Future Work

In this paper we introduced a new flooding algorithm employed on top of the cluster topology built by NLWCA. The protocol is simple and easy to deploy and aims to avoid high computation and communication load on the network nodes. The simulation results show that DWCF highly outperforms the inter-cluster broadcasting protocol of WCPD in terms of message spreading. On the other side, WCPD focus on the communication with nearby clusters considered to be stable. This is beneficial for applications which require that a multi-hop path between two communicating nodes has certain stability. In future work we aim to improve the DWCF by decreasing the delays on high weighted nodes in order

reduce the load on nodes with low battery power. Besides this, nodes with a higher number of neighboring foreign clusters should forward the message faster in order to decrease the network communication load.

References

1. Peleg, D.: Distributed computing: a locality-sensitive approach. Society for Industrial and Applied Mathematics Philadelphia, PA, USA (2000)
2. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H., Mit, C.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1(4), 660–670 (2002)
3. Dasgupta, K., Kalpakis, K., Namjoshi, P.: An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. Wireless Communications and Networking 3, 1948–1953 (2003)
4. Luo, H.G., Ye, F.G., Cheng, J.G., Lu, S.G., Zhang, L.G.: Ttdd: Two-tier data dissemination in large-scale wireless sensor networks. Wireless Networks 11(1), 161–175 (2003)
5. Zhu, S., Setia, S., Jajodia, S.: LEAP: Efficient security mechanisms for large- scale distributed sensor networks. In: Proceedings of the 10th ACM conference on Computer and communication security, Washington, DC, USA, pp. 62–72 (2003)
6. Andronache, A., Brust, M.R., Rothkugel, S.: Multimedia Content Distribution in Hybrid Wireless using Weighted Clustering. In: The 2nd ACM Workshop on Wireless Multimedia Networking and Performance Modeling. ACM Press, New York (2006)
7. Andronache, A., Brust, M., Rothkugel, S.: Hycast-podcast discovery in mobile networks. In: The Third ACM International Workshop on Wireless Multimedia Networking and Performance Modeling. ACM Press, New York (2007)
8. Andronache, A., Rothkugel, S.: NLWCA-Node and Link Weighted Clustering Algorithm for Backbone-assisted Mobile Ad Hoc Networks. In: The Seventh International Conference on Networking, Cancun, Mexico (2008)
9. Andronache, A., Rothkugel, S.: Hytrace-backbone-assisted path discovery in hybrid networks. In: International Conference on Communication Theory, Reliability, and Quality of Service, Bucharest, Romania (2008)
10. Hogie, L., Guinand, F., Bouvry, P.: A heuristic for efficient broadcasting in the metropolitan ad hoc network. In: 8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems, Wellington, New Zealand, pp. 727–733 (2004)
11. Gorgen, D., Frey, H., Hiedels, C.: JANE-The Java Ad Hoc Network Development Environment. In: 40th Annual Simulation Symposium (ANSS 2007), Norfolk, VA, pp. 163–176 (2007)
12. Ljubica, B., Silvia, G., Jean-Yves, L.B.: Self Organized Terminode Routing. Cluster Computing 5(2) (2002)
13. Alba, E., Dorronsoro, B., Luna, F., Nebro, A.J., Bouvry, P., Hogie, L.: A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitanmanets. Computer Communication Journal 30(4), 685–697 (2007)
14. Karlsson, G., Lenders, V., May, M.: Delay-Tolerant Broadcasting. In: SIGCOMM 2006 Workshops, ACM Press, New York (2006)
15. Sun, Y., Belding-Royer, E.M.: Application-oriented routing in hybrid wireless networks. In: IEEE International Conference on Communications, ICC 2003, Anchorage, Alaska, vol. 1, pp. 502–506 (2003)

Improved Time Complexities of Algorithms for the Directional Minimum Energy Broadcast Problem

Joanna Bauer* and Dag Haugland**

Department of Informatics, University of Bergen, PB. 7803, 5020 Bergen, Norway
{Joanna.Bauer,Dag.Haugland}@ii.uib.no

Abstract. Ability to find a low-energy broadcast routing quickly is vital to a wireless system's energy efficiency. Directional antennae save power by concentrating the transmission towards the intended destinations. A routing is given by assigning a transmission power, angle, and direction to every networking unit, and the problem of finding such a power saving routing is called the Directional Minimum Energy Broadcast Problem (D-MEBP). In the well known Minimum Energy Broadcast Problem (MEBP), the transmission angle is fixed to 2π. Previous works suggested to adapt MEBP algorithms to D-MEBP by two procedures, Reduced Beam (RB) and Directional (D). As the running time of the routing algorithms is a critical factor, we reduce the time complexity of both by one order of magnitude.

1 Introduction

In wireless ad-hoc networks, a broadcast session is established without use of any central backbone system, and is based entirely on message passing between network units. To accomplish this, each unit is equipped with an energy resource in terms of a battery. Since this resource is limited it becomes crucial to route the broadcast messages in such a way that power consumption is minimized. At each network unit transmitting a message, the power consumption typically depends on the transmission coverage, which in its turn is determined by the set of intended recipients.

What parameters that can be set in order to achieve a minimum energy broadcast routing depends on the technology of the transmission antennae in the network. In the case of directional antennae, the transmission beam is concentrated towards the intended destination units, and the coverage hence has both a radial and an angular dimension. The former is simply the power required to reach the most remotely located recipient, and the latter is the minimum angle of a sector containing all. For networks based on omnidirectional antennae, the transmission angle is fixed to 2π.

* Supported by L. Meltzers Høyskolefond under contract no. 480621.
** Corresponding author.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 488–496, 2008.

The Minimum Energy Broadcast Problem (MEBP) has in the case of omnidirectional antennae attracted intensive research. As the problem is NP-hard [1], the energy efficiency of applications depends on efficient routing heuristics. An overview of various suggestions to such methods can be found in the survey of Guo and Yang [2].

A common approach is to represent the network as a graph and determine a routing arborescence spanning the nodes in the graph. The arborescence defines an assignment of power to the nodes, given as the cost of the most expensive outgoing arc. A straightforward choice of routing arborescence is the Minimum Spanning Tree (MST), computed for instance by Prim's algorithm, which has been studied thoroughly. In [3], Guo and Yang proved that MST provides the optimal solution to a variant of MEBP, the static Maximum Lifetime Multicast Problem (MLMP).

Arborescences yielding a smaller total power assignment are found by taking into account the node-oriented objective function. In construction algorithms, where new nodes are added iteratively to an arborescence consisting initially of only the source node, this can be reflected by selecting nodes such that the incremental power is minimized. The most frequently cited such algorithm is the Broadcast Incremental Power (BIP) algorithm by Wieselthier et al. [4].

Assuming that the antennae are directional, we arrive at an extension of MEBP referred to as the Directional MEBP (D-MEBP). This problem has been studied to a far lesser extent than MEBP. Wieselthier et al. suggested in [5] the principles Reduced Beam (RB) and Directional (D) to adapt BIP to D-MEBP, resulting in the heuristics RB-BIP and D-BIP, respectively. RB-BIP first calls BIP to construct a broadcast routing arborescence, and then simply reduces the transmission angle of every unit to the minimum angle necessary to cover all the unit's children. D-BIP, on the other hand, takes antenna angles into account already in the construction phase. In each iteration of this procedure, the increase in both power requirement and angle are considered when deciding which unit to add to the current arborescence. In general, RB can be considered as a local improvement procedure to be called after construction, whereas D is interleaved with the construction algorithm.

In [6], Guo and Yang presented a mixed integer programming model, and used RB and D to adapt their local search heuristic [7] to D-MEBP. In [3], they applied both principles to adapt the MST algorithm to a directional version of MLMP.

As demonstrated in the above articles, RB and D are useful for adapting MEBP algorithms to D-MEBP in general. Wieselthier et al. showed in [8] that the time complexities of RB-BIP and D-BIP are $O(|V|^3)$ and $O(|V|^3 \log |V|)$, respectively, where V denotes the node set of the graph. The result for RB-BIP is derived from an implementation of BIP with $O(|V|^3)$ time complexity. The additional time complexity of the RB procedure is in [8] proved to be bounded by $O(|V|^2 \log |V|)$.

In this paper, we first improve the time complexity of RB to $O(|V|\log|V|)$ by better analysis. Together with an implementation of BIP with $O(|A|+|V|\log|V|)$ time complexity, suggested by Bauer et al. [9], this results in an implementation of RB-BIP with as low time complexity as $O(|A|+|V|\log|V|)$. Here A denotes the set of arcs in the graph.

Second, we suggest a novel implementation of D-BIP building on the BIP implementation in [9], and prove that its running time is $O(|V|^2)$.

2 Preliminaries

An instance of D-MEBP is given by a directed graph $G=(V,A)$, where the nodes represent the networking units, a source $s \in V$, power requirements $c \in \mathbb{R}^A$, and the minimum transmission angle $\theta_{\min}$. The nodes are associated with points in the plane, and the power requirement c_{vu} is typically proportional to d_{vu}^{α}, where d_{vu} is the Euclidean distance between nodes v and u, and $\alpha \in [2,4]$ is a constant [2].

A solution to any instance can be given by an s-arborescence $T=(V,A_T)$ with arc set $A_T \subseteq A$. An s-arborescence is a directed tree where all arcs are oriented away from s. In T, every node v has a (possibly empty) set $\Gamma_v(T)$ of children. The transmission power induced by T at $v \in V$ is given by

$$p_v(T) = \begin{cases} 0 & \text{if } \Gamma_v(T) = \emptyset \\ \max_{w \in \Gamma_v(T)}\{c_{vw}\} & \text{otherwise .} \end{cases}$$

In the idealized model assumed in the literature, the energy emitted by node v is concentrated uniformly in a beam of width $\theta_v(T)$ [5]. To simplify the definition of $\theta_v(T)$, nodes are identified with points in $\mathbb{R}^2$. For any two nodes u and v, we let uv denote the straight line segment in $\mathbb{R}^2$ with end points u and v, and for any three nodes u, v and w, we let $\angle_{uvw}$ denote the angle between the line segments uv and vw with positive (counter-clockwise) direction from uv to vw. This implies $\angle_{uvw} = 2\pi - \angle_{wvu}$. For the purpose of simplified presentation, we assume that no three nodes are collinear, and we define $\angle_{uvu} = 2\pi$. Let the *sector* S_{uvw} be defined as the node set $S_{uvw} = \{x \in V : \angle_{uvx} \le \angle_{uvw}\}$. For any node set $V' \subset V$, we define (see Fig. 1)

$$\theta_v(V') = \begin{cases} \theta_{\min} & \text{if } |V'| < 2 \\ \max\{\theta_{\min}, \min_{u,w \in V'}\{\angle_{uvw} : V' \subseteq S_{uvw}\}\} & \text{otherwise .} \end{cases} \quad (1)$$

The beam width $\theta_v(T)$ is hence given as $\theta_v(\Gamma_v(T))$. In Fig. 1, $t_v(T) \in \Gamma_v(T)$ and $t'_v(T) \in \Gamma_v(T)$ are the nodes for which the minimum in (1) is attained in the case $V' = \Gamma_v(T)$ and $|V'| \ge 2$.

The directional minimum energy broadcast problem can then be formulated as

[**D-MEBP**] Find an s-arborescence T such that $p_T = \sum_{v \in V} p_v(T)\theta_v(T)$ is minimized.

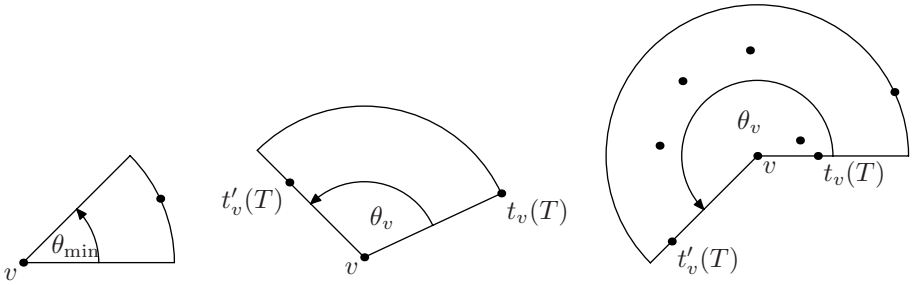


Fig. 1. Examples of beam width $\theta_v(T)$

3 The Reduced Beam Procedure

By (1) and the RB principle, any s-arborescence constructing heuristic $\mathcal{H}$ for MEBP can be extended to a D-MEBP heuristic RB-$\mathcal{H}$. This derived heuristic consists of the two steps $\mathcal{H}$ and the RB procedure. The latter simply amounts to computing $\theta_v(T)$ for all $v \in V$, which is accomplished by sorting the children of v according to the angular dimension of their polar coordinates with v as center.

By exploiting the fact that every node has at most $|V|$ children, Wieselthier et al. [5] found that the time complexity of sorting all children of all nodes is bounded by $O(|V|^2 \log |V|)$. However, since there are only a total of $|V| - 1$ children in the arborescence, the time complexity of sorting the children of all nodes is bounded by $O(|V| \log |V|)$. Thus we have the following result.

Theorem 1. *RB has $O(|V| \log |V|)$ time complexity.*

4 Directional BIP

The BIP-algorithm [4] for the omnidirectional version of the problem resembles Prim's algorithm for MST. In each iteration, the best arc from some connected node v to some disconnected node is selected. The algorithms are distinguished in that the selection criterion in BIP is not to minimize arc cost but rather *incremental* arc cost, that is, arc cost minus the cost of the most expensive arc leaving v selected so far.

In [5], the authors adapt BIP to the directional problem. The resulting algorithm is referred to as the Directional BIP (D-BIP) heuristic, which differs from BIP by taking antenna directions and beam widths into account when selecting the next node to be added to the arborescence. In the implementation of D-BIP suggested in [8], the children of every node are maintained as sorted lists. The authors prove that the time complexity of such an implementation is bounded above by $O(|V|^3 \log |V|)$. Through computational experiments, it is also demonstrated that D-BIP outperforms RB-BIP for a large number of test instances.

In the following, we present an implementation of D-BIP that has $O(|V|^2)$ time complexity. It is presented as an extension of BIP, which in its turn can be seen as an extension of Prim's algorithm for MST. Then we demonstrate that these extensions are accomplished without distortion of the quadratic time complexity known to hold for Prim's algorithm.

4.1 An Implementation of BIP with Quadratic Running Time

Consider the $O(|A| + |V| \log |V|)$ implementation of Prim's algorithm shown in Table 1, based on the implementation given in [10]. The excluded nodes $V \setminus V_T$ are stored in a priority queue Q. We denote the key value of node v in Q by $\texttt{key}_Q[v]$. The operation Q.`extractMin`() and Q.`extractMax`() remove a node with smallest and largest key value, respectively, and return the removed node to the invoking algorithm. An array `parent` is maintained such that for all $v \in V \setminus V_T$, `parent`$[v]$ is the best parent node of v in V_T.

In all algorithms to follow, we assume that the graph G is represented by a set of adjacency lists $\{\texttt{Adj}[v] : v \in V\}$, where $\texttt{Adj}[v] = \{u : (v, u) \in A\}$.

Table 1. Prim's Algorithm

```
Prim(G = (V, A), s, c)
1   T = (V_T, A_T) ← ({s}, ∅)
2   priority queue Q ← V \ {s}
3   for all v ∈ Q
4       parent[v] ← s
5       key_Q[v] ← c_sv
6   while Q ≠ ∅
7       w ← Q.extractMin()
8       v ← parent[w]
9       V_T ← V_T ∪ {w}
10      A_T ← A_T ∪ {(v, w)}
11      for all u ∈ Adj[w]
12          if u ∈ Q ∧ c_wu < key_Q[u]
13              key_Q[u] ← c_wu
14              parent[u] ← w
15  return T
```

Table 2. Additional steps needed to extend Prim's algorithm to BIP

```
1       for all u ∈ Adj[v]
2           if u ∈ Q ∧ c_vu − c_vw < key_Q[u]
3               key_Q[u] ← c_vu − c_vw
4               parent[u] ← v
```

Assume the steps in Table 2 are inserted after the for-loop occupying Steps 11-14 in Table 1. In [9], it is proved that this extension results in an implementation of BIP with running time $O(|A| + |V| \log |V|)$.

4.2 Directional BIP as an Extension of Prim's Algorithm

Consider a tree $T = (V_T, A_T)$ where $V_T \subset V$, and a node $v \in V_T$ for which $\Gamma_v(T) \neq \emptyset$. Since no three nodes in $\Gamma_v(T)$ are collinear, there exists for each node $u \in \Gamma_v(T)$ a unique $u' \in \Gamma_v(T)$ such that $S_{uvu'} \cap \Gamma_v(T) = \{u, u'\}$. With reference to a polar coordinate system centered at v, u' is the successor of u when sorting $\Gamma_v(T)$ by increasing value of the angular dimension (defined cyclically such that u' is the first node if u is the last). If $|\Gamma_v(T)| = 1$, then $u' = u$.

Define the family of sectors hence induced by node v as $\mathcal{S}_v(T) = \{S_{uvu'} : u \in \Gamma_v(T)\}$. In the example shown in Fig. 2, the sectors induced by v are S_{wvx}, S_{xvy}, S_{yvz} and S_{zvw}.

The figure also illustrates the general fact that $\theta_v(T) = \max\{\theta_{\min}, 2\pi - \max\{\angle_{uvu'} : u \in \Gamma_v(T)\}\}$, which means that if $\theta_v(T) > \theta_{\min}$, then the complementary angle of $\theta_v(T)$ is the angle of a widest sector in $\mathcal{S}_v(T)$. This observation is used to maintain information on the incremental cost of adding a new arc to T.

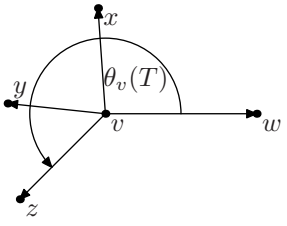


Fig. 2. Sectors induced by v and $\Gamma_v(T)$

For all $u \in V \setminus V_T$ such that $(v, u) \in A$, we need to know the new value of $\theta_v(T)$ given that (v, u) is added to A_T. Consider the case where $|\Gamma_v(T)| > 1$, and let $S_{zvz'}$ and $S_{yvy'}$ be the two widest sectors in $\mathcal{S}_v(T)$ (ties broken arbitrarily), where $\angle_{zvz'} \geq \angle_{yvy'}$. When evaluating the inclusion of u in V_T, we have to take into account how u relates to $S_{zvz'}$ and $S_{yvy'}$:

- If $u \notin S_{zvz'}$, then $S_{zvz'}$ will remain the widest sector in $\mathcal{S}_v(T)$ if (v, u) is added to A_T, and thus $\theta_v(T)$ is unchanged.
- If $u \in S_{zvz'}$, then adding (v, u) to A_T implies that $S_{zvz'}$ leaves $\mathcal{S}_v(T)$, whereas S_{zvu} and $S_{uvz'}$ enter. Consequently, the widest sector in the updated family $\mathcal{S}_v(T)$ is S_{zvu}, $S_{uvz'}$ or $S_{yvy'}$. The new value of $\theta_v(T)$ thus becomes $\max\{\theta_{\min}, 2\pi - \max\{\angle_{zvu}, \angle_{uvz'}, \angle_{yvy'}\}\}$.

It follows that access to the two widest sectors in $\mathcal{S}_v(T)$ is crucial for rapid computation of the incremental cost of adding a potential new arc to A_T. In our implementation of D-BIP, we therefore represent $\mathcal{S}_v(T)$ by a priority queue $\mathcal{S}_v$ where $S_{uvu'}$ has key value $\texttt{key}_{\mathcal{S}_v}[(u, u')] = \angle_{uvu'}$. The methods $\mathcal{S}_v.\texttt{insert}\big((u, u')\big)$ and $\mathcal{S}_v.\texttt{delete}\big((u, u')\big)$ are used to add/remove sector $S_{uvu'}$ to/from the queue.

Table 3 shows the steps that replace Steps 11-14 in Table 1 when extending Prim's algorithm to D-BIP. We make use of a matrix $\theta \in \mathbb{R}^A$, where θ_{vu} is the value of $\theta_v(T)$ resulting from the possible inclusion of arc (v, u) in A_T.

To complete the extension of Prim's algorithm to D-BIP, Step 5 in Table 1 has to be changed to

5 $\quad \texttt{key}_Q[v] \leftarrow c_{sv}\theta_{\min}$,

Table 3. Additional steps needed to extend Prim's algorithm to D-BIP

```
1   for all u ∈ Adj[v] ∩ Q
2       if c_wu θ_min < key_Q[u]
3           key_Q[u] ← c_wu θ_min
4           parent[u] ← w
5   if Γ_v(T) = {w}
6       priority queue S_v ← {(w, w)}
7       key_{S_v}[(w, w)] ← ∠_vww
8       for all u ∈ Q ∩ Adj[v]
9           θ_vu ← max{θ_min, min{∠_uvw, ∠_wvu}}
10  else
11      find (x, x') ∈ S_v : w ∈ S_xvx'
12      S_v.delete((x, x'))
13      key_{S_v}[(x, w)] ← ∠_xvw, S_v.insert((x, w))
14      key_{S_v}[(w, x')] ← ∠_wvx', S_v.insert((w, x'))
15      (z, z') ← S_v.extractMax(), (y, y') ← S_v.extractMax()
16      S_v.insert((y, y')), S_v.insert(z, z'))
17      for all u ∈ Q ∩ Adj[v]
18          if u ∈ S_zvz'
19              θ_vu ← max{θ_min, 2π − max{∠_zvu, ∠_uvz', ∠_yvy'}}
20          else
21              θ_vu ← max{θ_min, ∠_zvz'}
22  for all u ∈ Adj[v] ∩ Q
23      incCost ← max{p_v(T), c_vu}θ_vu − p_v(T)θ_v(T)
24      if incCost < key_Q[u]
25          key_Q[u] ← incCost
26          parent[u] ← v
```

in order to reflect that the cost of adding arc (s, v) is $c_{sv}\theta_{\min}$ rather than c_{sv}. Accordingly, Steps 1-4 in Table 3 are updates of Steps 11-14 in Table 1, taking the minimum beam width into account.

Steps 5-21 concern the updates of $\mathcal{S}_v$ and θ_{vu} after insertion of (v, u) in A_T. Steps 22-26 correspond to the extension made for BIP (Table 2), except that power has been replaced by power times beam width.

Theorem 2. *D-BIP has $O(|V|^2)$ time complexity.*

Proof. All the steps in Table 3 are included in the while-loop in Table 1, which generates $|V|$ iterations. We therefore need to show that each of these steps has at most $O(|V|)$ time complexity.

For the for-loop 1-4, this follows from the analysis of Prim's algorithm.

Given that the priority queues are implemented as Fibonacci heaps, the insertion and key update operations run in constant amortized time, and the operations `delete` and `extractMax` run in $O(\log n)$ amortized time, where n is the maximum number of elements in the queue. Furthermore, any angle $\angle_{uvw}$

is computed in constant time, and checking whether $u \in S_{zvz'}$ is also done in constant time. Thus, each step within the for-loops 8-9, 17-21 and 22-26 runs in constant (amortized) time, and the loops generate at most $|V|$ iterations each. Furthermore, Steps 6-7 and 12-16 have constant and $O(\log |V|)$ time complexity, respectively.

The proof is complete by observing that Step 11 has time complexity $O(|V|)$ since $|\mathcal{S}_v| \leq |V|$. □

5 Conclusions

We have studied how to extend construction heuristics designed for the Minimum Energy Broadcast Problem to the directional version of the problem. Two approaches from the literature, RB and D, were chosen, and we have given fast implementations of both. By virtue of the implementations and analysis given in the current work, the time complexities of previously suggested methods like RB-BIP and D-BIP are reduced by one order of magnitude.

This achievement can be generalized in several directions. Due to the generality of RB and D, our results can be transferred also to other existing and future construction heuristics for MEBP. To simplify the presentation, we have chosen to present the implementations for broadcast routing, but they can easily be adapted to the more general multicast case.

References

1. Clementi, A.E.F., Crescenzi, P., Penna, P., Rossi, G., Vocca, P.: On the Complexity of Computing Minimum Energy Consumption Broadcast Subgraphs. In: Ferreira, A., Reichel, H. (eds.) STACS 2001. LNCS, vol. 2010, pp. 121–131. Springer, Heidelberg (2001)
2. Guo, S., Yang, O.: Energy-aware Multicasting in Wireless Ad Hoc Networks: A Survey and Discussion. Computer Communications 30(9), 2129–2148 (2007)
3. Guo, S., Yang, O.: Multicast Lifetime Maximization for Energy-constrained Wireless Ad-hoc Networks with Directional Antennas. In: Proceedings of the Globecom 2004 Conference, pp. 4120–4124. IEEE Computer Society Press, Dallas (2004)
4. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. ACM Mobile Networks and Applications Journal 7(6), 481–492 (2002)
5. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: Energy-limited Wireless Networking with Directional Antennas: The Case of Session-based Multicasting. In: Proceedings IEEE INFOCOM 2002, pp. 190–199. IEEE Press, New York (2002)
6. Guo, S., Yang, O.: Minimum-energy Multicast in Wireless Ad Hoc Networks with Adaptive Antennas: MILP Formulations and Heuristic Algorithms. IEEE Transactions on Mobile Computing 5(4), 333–346 (2006)
7. Guo, S., Yang, O.: A Dynamic Multicast Tree Reconstruction Algorithm for Minimum-energy Multicasting in Wireless Ad Hoc Networks. In: Hassanein, H., Oliver, R.L., Richard III, G.G., Wilson, L.F. (eds.) 23rd IEEE International Performance, Computing, and Communications Conference, pp. 637–642. IEEE Computer Society, Los Alamitos (2004)

8. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: Energy-aware Wireless Networking with Directional Antennas: The Case of Session-based Broadcasting and Multicasting. IEEE Transactions on Mobile Computing 1(3), 176–191 (2002)
9. New Results on the Time Complexity and Approximation Ratio of the Broadcast Incremental Power Algorithm. Under review (2008)
10. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2001)

Optimised Recovery with a Coordinated Checkpoint/Rollback Protocol for Domain Decomposition Applications

Xavier Besseron and Thierry Gautier

MOAIS Project
Laboratoire d'Informatique de Grenoble
ENSIMAG - Antenne de Montbonnot
ZIRST 51, avenue Jean Kuntzmann
38330 Montbonnot Saint Martin, France
{xavier.besseron,thierry.gautier}@imag.fr

Abstract. Fault-tolerance protocols play an important role in today long runtime scientific parallel applications. The probability of a failure may be important due to the number of unreliable components involved during an execution. In this paper we present our approach and preliminary results about a new checkpoint/rollback protocol based on a coordinated scheme. One feature of this protocol is that fault recovery only requires a partial restart of other processes thanks to the availability of an abstract representation of the execution. Simulations on a domain decomposition application show that the amount of computations required to restart and the number of involved processes are reduced compared to the classical global rollback protocol.

Keywords: grid, fault-tolerance, parallel computing, data flow graph.

1 Introduction

Since few years, fault-tolerance has been studied in the context of high-performance parallel applications that makes use of large scale clusters or grids (i.e. simulation of complex phenomena) [1,2,3,4,5,6]. Due to the number of unreliable components involved during the computation, the apparition of faults is not an exceptional event [7,8]: the system or the middleware should provide fault-tolerance protocols in order to mask failures. Moreover, some applications require an important computation time to complete (like a week running on a thousand processors [9]). Exclusive reservation of computing resources during such a period conflicts with reservation policies aiming at fairness between users on short periods. In this case, fault-tolerance allows to split a large computation and run it during many shorter separated reservations [10,11].

This subject has been well studied in the context of distributed systems and distributed middlewares [1,2,12,13]. Optimising performance on large scale architectures becomes a major objective. Recent propositions study the applications

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 497–506, 2008.

runtime behaviour in order to specialise or extend published protocols [5,6,14,15]. This is the context of our paper.

In our document the specialisation of fault-tolerance protocol is done using an abstract representation of the execution offering important optimisations at runtime. We implemented this work in the framework of Kaapi [4,14,16], where the abstract representation of execution was firstly designed to plug scheduling algorithms independently of applications. In [4,6], it was shown that this abstract representation is well suited for defining the local process checkpoint. In this paper, this abstract representation is used to specialise a fault-tolerance protocol for long runtime intensive iterative simulation where the communications versus computing ratio is high.

Experiments carried out in [2,5] show that coordinated checkpoint/rollback protocols are efficient up to thousands of processors. In case of fault, all the processors restart from their most recent checkpoint, even those which did not failed. The two challenging problems about performances of coordinated checkpoint/rollback protocols are:

1. How to speed up processes restart after the occurrence of a fault?
2. How to reduce the amount of computation time loss in case of fault?

In [2,5] the solution to solve (1) is: each process keeps a local copy of its checkpoint and sends another copy to either a stable storage [5] or a fixed number of neighbour processes [2]. Within this approach, all processes except the failed process, restart from their local copy of the most recent checkpoint.

Our contribution is mainly to propose a solution for (2). Thanks to the abstract representation of execution of any Kaapi applications, it is possible to compute the strictly required computation set which is the computation task set that a processor have to re-execute to resend lost messages to the failed processor. This optimisation reduces the amount of computation required to restart the application. Furthermore, if the task set is parallel enough, it can be scheduled over all the processors to speed up the restart.

The outline of the paper is the following. The next section deals with related works. Section three presents the improved rollback of our coordinated checkpoint/rollback protocol. It begins with an overview of the abstract representation in Kaapi and the process state definition. Then we present the recovery step and an analysis of its complexity is sketched. The next section presents a study case on a domain decomposition application and simulations of its restart. The conclusion ends the paper.

2 Related Works

In this paper we deal with long runtime of parallel applications with a high ratio communication versus computation. Such kind of applications appear during iterative simulation of physical phenomena: for instance molecular dynamics [17], virtual reality [18]. Parallelisation of such applications uses domain decomposition method: the simulation domain is splitted into smaller subdomains. During

an iteration, each processor communicates with its neighbours according to subdomain relationship.

Fault-tolerance protocols have been classified in three categories [1]: those based on duplication to introduce redundancy of computations [12,19]; protocols based on event logging [20] and protocols based on checkpoint/rollback approach [1,21].

Protocols based on duplication only tolerate a fixed number of faults and may consume lots of resources [19]. Since the main criteria for the considered applications is the performance, and moreover, an interruption during the computation can be tolerated, they are not selected.

Log-based protocols assume that the state of the system evolves according to non-deterministic events. These events are logged in order to rollback from a previous saved checkpoint [1]. In our case, non-deterministic events are communications between subdomains which represent a large amount of data. So these protocols are not selected, they require too many resources (memory, bandwidth) [3].

Checkpoint/rollback protocols periodically save the local process state of the applications and have few overhead with respect to the communications. They come in three forms depending on the way they build a coherent global state for the application restart [1]. Uncoordinated protocols make no assumption about the coherency of the global state captured and may be impacted by the domino effect: in worst case, the application is required to rollback at the beginning [22]. Coordinated protocols are based on global synchronisation to ensure that the set of local checkpoints forms a coherent global state [21]. Communication-induced checkpointing protocols [23] are a mix between coordinated and uncoordinated protocols where forced checkpoints are computed on reception of some messages.

Coordinated checkpoint/rollback protocols have the advantage of having a low overhead towards application communications [2,5]. However, they produce a large communication volume due to the checkpoints size which are sent simultaneously to the checkpoints servers. This can be amortised by choosing a suitable checkpoint period [3] or using incremental checkpoints [24].

3 Improved Coordinated Checkpoint/Rollback Protocol

The idea of the Coordinated Checkpointing in Kaapi (CCK) protocol is to build after fault occurrence, the computations of every processes that are strictly required to resend messages to the failed processor. Thanks to Kaapi, the amount of computation to re-execute is less than in classical and improved coordinated protocols [1,2,5] for which all the processors restart from their last checkpoint.

This section presents how to reduce the number of instruction to re-execute using the execution abstract representation provided by Kaapi. We first describe the execution model and the abstract representation of Kaapi. Then we deal with the optimised recovery.

3.1 Execution Model and Abstract Representation

KAAPI [1] [16] is a middleware that allows to execute distributed and/or parallel applications. It offers a high level parallel programming model. The programmer writes his program describing potential parallelism independently of the target architecture, using for example the ATHAPASCAN [25,26] programming interface.

With ATHAPASCAN, the parallelism is defined with two simple concepts: *shared data* and *tasks*. A shared data is a data in global memory that a task can produce or consume. A task is an indivisible instruction set that declares an access mode to a shared data (read or write). With this description, KAAPI can execute the application according to the *precedence constraints* which are dynamically detected.

The set {tasks, shared data, precedence constraints} builds the data flow graph representing the application execution [26]. A data flow graph is defined as a directed graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite vertex set (tasks and shared data) and $\mathcal{E}$ is an edge set (precedence constraints) between vertices. This data flow graph is called the *abstract representation* of the application. This representation is causally connected to the (execution of the) application: any new execution of an API instruction is reported by the creation of new vertices in the data flow graph; and any modification in the data flow graph is rendered in a modification in the application execution. For instance, the data flow graph is distributed among the processes and the application execution reflects this by having (generally) speedup in comparison to the sequential execution.

For the application aimed in this paper, we use the following approach, called *static scheduling* [11], to execute the data flow graph. First, a pluggable library like SCOTCH [27] or METIS [28] partitions the data flow graph of one iteration in N data flow subgraphs where N is the wanted processor number to run on. For each subgraph, the static scheduling KAAPI engine automatically generates the tasks for the required communications. Then data flow subgraphs are distributed over all the processors and they execute their subgraph iteratively. If no modification of the data flow graph occurs between iterations then subgraphs are reused without recomputing them.

3.2 Definition of a Checkpoint

The application state is represented by the state of all its processes and by the state of communication channels. Because the communication channels' state is not available, the principle of coordinated protocols is the synchronise all the processes and to flush all in-transit messages in order to checkpoint the application. Under this condition, the application state is made of the union of all the process local states [21].

The process state can be save using its abstract representation as a data flow graph G_i (which is composed of the graph and its input data). Moreover, this state is independent of the computer executing the process (hardware, operating system) if it is saved between the execution of two tasks.

[1] `http://kaapi.gforge.inria.fr`

Definition 1. *The checkpoint G_i of process P_i is composed of its data flow graph, i.e. its tasks and their associated inputs. It does not depend on the task execution state on the processor itself.*

Finally, a coherent global state G of the application is the set of all the local checkpoints G_i which are saved during the same coordination step.

The checkpointing step of CCK protocol implemented in KAAPI is based on the classical coordinated checkpointing protocol presented in [21] and on optimisations proposed in [29]. It is fully detailled in [30,31].

3.3 Recovery After Failures

When one or many processes fail during the computation, the role of a checkpoint/rollback protocol is to restart the application in a state that could happen in a normal execution (i.e. without failure). At the failure time, the application is composed of two kind of processes: failed processes and non-failed processes. The last checkpoint of all processes is available and all these checkpoints form a coherent global state of the application before the failure. Furthermore, the current state of the non-failed processes is known.

In the case of the classical rollback protocol [21], all processes would restart from their last checkpoint (failed processors are replaced using spare processors). However, all computations performed on all the processes since the last checkpoint step are lost. This waste can be important specially when a large processor number is used.

The CCK rollback protocol try to reduce this waste. The substituting processes that replace failed processes have to restart from the last checkpoint because the failure made failed processes loose their current state. As for the non-failed processes, they keep their ongoing computation. Because the global state made of the states of substituting processes and non-failed processes is not coherent, the computation can't continue from this state. Analysing the execution abstract representation as a data flow graph allow us to identify, among the last checkpoint of non-failed processes, the *strictly required computation set* that need to be re-executed so as to guarantee that this global state is coherent.

Definition 2. *The **strictly required computation set** for a process P_i with respect to a process P_k is the minimal task set stored in the previous checkpoint of P_i which have already been executed on P_i and which produce, directly or indirectly, a data that will be send to P_k.*

The distributed algorithm that determines the strictly required computation set to re-execute is detailed in [31]. This algorithm computes the task set which produces data that will be send to failed processes by analysing the data flow graph stored in the previous checkpoint of each process. The demonstration that all lost messages is re-send is based on the properties of KAAPI execution model and data flow graph's. The coordination flushes all in-transit messages which imply that the set of local checkpoints is a coherent global state; so if a failed process P_{failed} should have received a message from process P_i, then there is a task in P_i that will produce the data consumed by task in P_{failed}.

3.4 Complexity Analysis

In this section we analyse the execution complexity with a fault in comparison to the complexity of classical coordinated checkpoint protocol [21] that restart all processes when one is faulty.

The worst case for our protocol is the case where the strictly required computation set of P_i with respect to P_{failed} contains all executed tasks on P_i. If it is true for all processes P_i, then our protocol's complexity is the traditional protocols' complexity plus the complexity to analyse the data flow graph in order to compute the strictly required computation set. This latter complexity corresponds to the computation of transitive closures on the graphs, which is linear with respect to the task number in the data flow graph to analyse because they are acyclic and directed [32].

Nevertheless, for the considered class of parallel applications, our algorithm's complexity is lesser than the classical coordinated protocol on two points:

1. The number of involved processes in the restart of P_{failed} is less that the total number of processes that have to restart for the classical protocol. Moreover, this number may be a constant.
2. The task number in the strictly required computation set is generally less than the executed tasks.

The point 1 is due to the fact that the knowledge of the data flow graph permits to know the communications between processes. The point 2 is due to the nature of the dependencies on some applications, especially in domain decomposition applications that exhibit good locality of (remote) data accesses because most of the computations use data from the process itself, only a few computations require data from other processes. These processes are bordering processes (according to subdomain relationship) and are in constant number.

4 Simulations

In this section we present simulations of the recovery step of CCK after one process failed. We consider an application that solves the Poisson problem on a large domain to study gains with a large processor number. The application uses the Jacobi method on a three dimensional domain. The size of the domain is $2,048^3$ (64 GB) split in 64^3 subdomains of 32 KB size. For each computation iteration, a subdomain update corresponds to one computation task. The execution of this task requires the neighbour subdomains. On a reference computer (Bi-Opteron 2 GHz CPU with a 2 GB memory), the execution of one computation task lasts 10 ms.

4.1 Checkpoint Period Influence

For this simulation, the 64^3 submains are distributed on 1,024 processors, so there are 256 subdomains (64 MB) for each process. In this case, the execution of one iteration (i.e. the update of all the subdomains) last about 2.5 seconds.

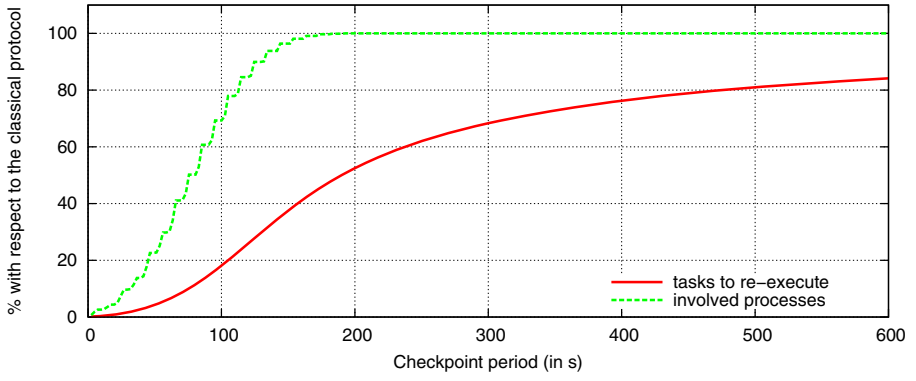


Fig. 1. Proportion in the worst case of tasks to re-execute and of involved processes for CCK restart with respect to the classical protocol

The figure 1 shows the proportion, with respect to the classical protocol, of tasks to re-execute and of involved processes for the CCK restart in relation to the checkpoint period. The curve shows the worst case values, i.e. when the failure happens just before the next checkpoint. With a 60-second period, less than 30 % of the processes are involved and only 6 % of the tasks have to be re-executed with respect to the classical protocol.

In order to reduce the restart time, the task set to re-execute can be distributed on all the processors. In this case, the estimated restart time is 3.6 seconds for CCK instead of 60 seconds for the global restart of the classical protocol. To this time, we have to add the time to identify and to distribute the strictly required task set. These will be evaluated in future experimentations.

4.2 Processor Number Influence

The two next simulations show the processor number influence on the CCK restart. The figure 2 shows the proportion of tasks to re-execute in comparison with the global restart for many checkpoint periods. On the figure 3 is the number of involved processes. For the scenario application run on 8,192 processors, a 10-seconds checkpoint period gives less than 10 % of tasks to re-execute and less than 2,500 involved processes (over 8,192).

Between two checkpoints, the amount of computation and the iteration number increase proportionally with the processor number. When the processor number increases, the proportion of tasks to re-execute and the number of involved processes also increase because the application graph is bigger and holds more dependencies. To preserve the protocol performances, it is required to decrease the checkpoint period when the processor number increases. Moreover, it guarantee that in case of failure, the lost computation will not be too big [3].

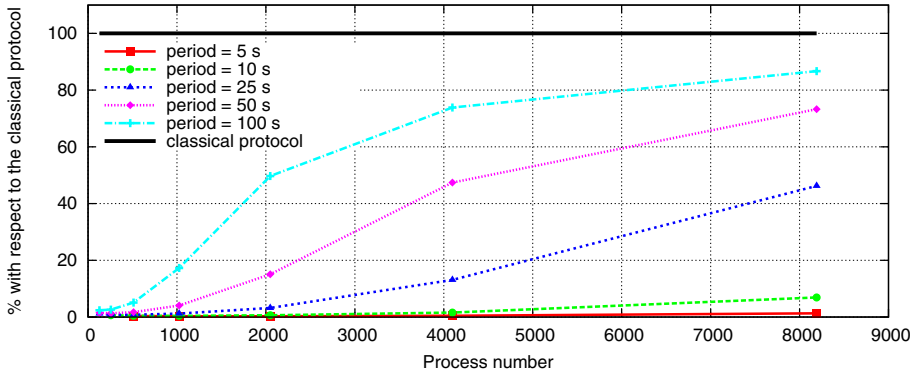


Fig. 2. Proportion of tasks to re-execute with CCK restart with respect to the classical protocol for many checkpoint periods in relation to the process number

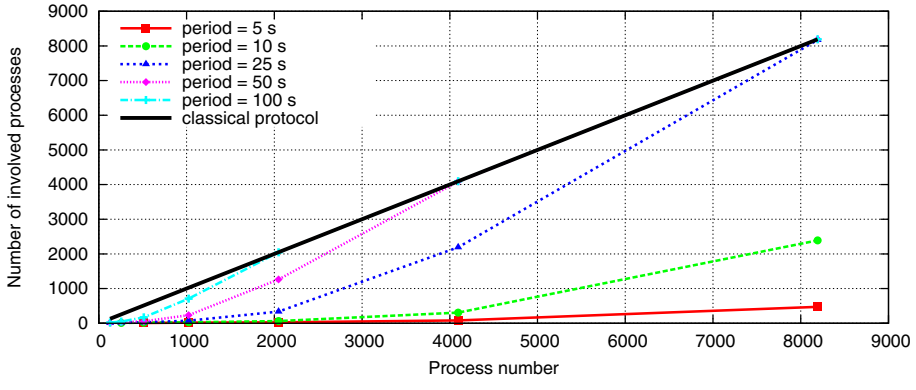


Fig. 3. Number of involved processes with CCK restart with respect to the classical protocol for many checkpoint periods in relation to the process number

5 Conclusion

In this paper we presented the CCK protocol, an improved coordinated checkpoint/rollback protocol for parallel applications. Our work originality comes from the abstract representation provided by the Kaapi library for any applications' parallel execution. The main contribution is to show how to improve classical coordinated checkpoint protocol by using a better knowledge of the application and especially about the dependencies between processes due to communications. We improved the application restart after failure: 1/ the number of processes involved in the restart is smaller; 2/ the restart time for this partial restart is shorter. This work is still in progress, additional evaluations and experiments at grid scale are planned. The final purpose is to provide a framework that adapts dynamically to available resources [11] using the CCK fault-tolerance protocol.

References

1. Elnozahy, E.N.M., Alvisi, L., Wang, Y.M., Johnson, D.B.: A survey of rollback-recovery protocols in message-passing systems. ACM Comput. Surv. 34(3), 375–408 (2002)
2. Zheng, G., Shi, L., Kalé, L.V.: Ftc-charm++: An in-memory checkpoint-based fault tolerant runtime for charm++ and MPI. In: 2004 IEEE International Conference on Cluster Computing, San Dieago, CA (September 2004)
3. Elnozahy, E.N., Plank, J.S.: Checkpointing for peta-scale systems: A look into the future of practical rollback-recovery. IEEE Transactions on Dependable and Secure Computing 1(2), 97–108 (2004)
4. Jafar, S., Krings, A.W., Gautier, T., Roch, J.L.: Theft-induced checkpointing for reconfigurable dataflow applications. In: IEEE, (ed.): IEEE Electro/Information Technology Conference (EIT, Lincoln, Nebraska (May 2005) This paper received the EIT 2005 Best Paper Award
5. Bouteiller, A., Lemarinier, P., Krawezik, G., Cappello, F.: Coordinated checkpoint versus message log for fault tolerant MPI. In: Proceedings of The 2003 IEEE International Conference on Cluster Computing, Honk Hong,China (2003)
6. Jafar, S., Krings, A., Gautier, T.: Flexible Rollback Recovery in Dynamic Heterogeneous Grid Computing. IEEE Transactions on Dependable and Secure Computing (TDSC) (in print, 2008)
7. Xie, M., Dai, Y.S., Poh, K.L.: Reliability of Grid Computing Systems. In: Computing System Reliability, pp. 179–205. Springer, US (2004)
8. Neokleous, K., Dikaiakos, M., Fragopoulou, P., Markatos, E.: Grid reliability: A study of failures on the egee infrastructure. In: Gorlatch, S., Bubak, M., Priol, T. (eds.) Proceedings of the CoreGRID Integration Workshop 2006, pp. 165–176 (Octobert 2006)
9. Anstreicher, K.M., Brixius, N.W., Goux, J.P., Linderoth, J.: Solving large quadratic assignment problems on computational grids. Technical report, Iowa City, Iowa 52242 (2000)
10. Wang, Y.M., Huang, Y., Vo, K.P., Chung, P.Y., Kintala, C.: Checkpointing and its applications. In: Fault-Tolerant Computing, 1995. FTCS-25. Digest of Papers, Twenty-Fifth International Symposium on (27-30 Jun 1995), pp. 22–31 (1995)
11. Jafar, S., Pigeon, L., Gautier, T., Roch, J.L.: Self-adaptation of parallel applications in heterogeneous and dynamic architectures. In: IEEE, (ed.): ICTTA 2006, IEEE Conference on Information and Communication Technologies: from Theory to Applications, Damascus, Syria, pp. 3347–3352 (April 2006)
12. Avizienis, A.: Fault-tolerant systems. IEEE Trans. Computers 25(12), 1304–1312 (1976)
13. Bosilca, G., Bouteiller, A., Cappello, F., Djilali, S., Fédak, G., Germain, C., Hérault, T., Lemarinier, P., Lodygensky, O., Magniette, F., Néri, V., Selikhov, A.: Mpich-v: Toward a scalable fault tolerant mpi for volatile nodes. In: SuperComputing, Baltimore, USA (2002)
14. Jafar, S., Gautier, T., Krings, A.W., Roch, J.-L.: A checkpoint/recovery model for heterogeneous dataflow computations using work-stealing. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, pp. 675–684. Springer, Heidelberg (2005)
15. Baude, F., Caromel, D., Delbé, C., Henrio, L.: A hybrid message logging-cic protocol for constrained checkpointability. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, pp. 644–653. Springer, Heidelberg (2005)

16. Gautier, T., Besseron, X., Pigeon, L.: Kaapi: A thread scheduling runtime system for data flow computations on cluster of multi-processors. In: PASCO 2007: Proceedings of the 2007 international workshop on Parallel symbolic computation, pp. 15–23 (2007)
17. Kal, L., Skeel, R., Bhandarkar, M., Brunner, R., Gursoy, A., Krawetz, N., Phillips, J., Shinozaki, A., Varadarajan, K., Schulten, K.: Namd2: greater scalability for parallel molecular dynamics. J. Comput. Phys. 151(1), 283–312 (1999)
18. Revire, R., Zara, F., Gautier, T.: Efficient and easy parallel implementation of large numerical simulation. In: Dongarra, J., Laforenza, D., Orlando, S. (eds.) EuroPVM/MPI 2003. LNCS, vol. 2840, pp. 663–666. Springer, Heidelberg (2003)
19. Wiesmann, M., Pedone, F., Schiper, A.: A systematic classification of replicated database protocols based on atomic broadcast. In: Proceedings of the 3rd Europeean Research Seminar on Advances in Distributed Systems (ERSADS 1999), Madeira Island, Portugal (1999)
20. Alvisi, L., Marzullo, K.: Message logging: Pessimistic, optimistic, causal, and optimal. IEEE Transactions on Software Engineering 24(2), 149–159 (1998)
21. Chandy, K.M., Lamport, L.: Distributed snapshots: determining global states of distributed systems. ACM Trans. Comput. Syst. 3(1), 63–75 (1985)
22. Randell, B.: System structure for software fault tolerance. In: Proceedings of the international conference on Reliable software, pp. 437–449 (1975)
23. Baldoni, R.: A communication-induced checkpointing protocol that ensures rollback-dependency trackability. In: Proc. of the 27th International Symposium on Fault-Tolerant Computing (FTCS 1997), p. 68. IEEE Computer Society, Los Alamitos (1997)
24. Plank, J.S., Beck, M., Kingsley, G., Li, K.: Libckpt: Transparent Checkpointing under Unix. In: Proceedings of USENIX Winter 1995 Technical Conference, New Orleans, Louisiana,USA, pp. 213–224 (January 1995)
25. Galilée, F., Roch, J.L., Cavalheiro, G., Doreille, M.: Athapascan-1: On-line building data flow graph in a parallel language. In: IEEE, (ed.): Pact 1998, Paris, France, pp. 88–95 (October 1998)
26. Roch, J.L., Gautier, T., Revire, R.: Athapascan: Api for asynchronous parallel programming. Technical Report RT-0276, Projet APACHE, INRIA (February 2003)
27. Pellegrini, F., Roman, J.: Experimental analysis of the dual recursive bipartitioning algorithm for static mapping. Technical Report 1038-96, LaBRI, Université Bordeaux I (1996)
28. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: Application in VLSI domain. In: Proceedings of the 34th annual conference on Design automation, pp. 526–529. ACM Press, New York (1997)
29. Koo, R., Toueg, S.: Checkpointing and rollback-recovery for distributed systems. IEEE Trans. Softw. Eng. 13(1), 23–31 (1987)
30. Besseron, X., Jafar, S., Gautier, T., Roch, J.L.: Cck: An improved coordinated checkpoint/rollback protocol for dataflow applications in kaapi. In: IEEE, (ed.): ICTTA 2006, IEEE Conference on Information and Communication Technologies: from Theory to Applications, Damascus, Syria, pp. 3353–3358 (April 2006)
31. Besseron, X., Pigeon, L., Gautier, T., Jafar, S.: Un protocole de sauvegarde / reprise coordonné pour les applications à flot de données reconfigurables. Technique et Science Informatiques - numéro spécial RenPar"17 27 (2008)
32. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)

A Context-Aware Broadcast Protocol for Mobile Wireless Networks

Luc Hogie[1], Grégoire Danoy[2], Pascal Bouvry[2], and Frédéric Guinand[3]

[1] MASCOTTE, joint project CNRS-INRIA-UNSA*
2004, route des lucioles - BP 93
FR-06902 Sophia Antipolis Cedex
luc.hogie@sophia.inria.fr
[2] University of Luxembourg
6, rue R. Coudenhove-Kalergi
L-1359 Luxembourg
gregoire.danoy@uni.lu,
pascal.bouvry@uni.lu
[3] University of Le Havre
25, rue Philippe Lebon, BP 1123
FR-76063 Le Havre Cedex
frederic.guinand@univ-lehavre.fr

Abstract. Delay Tolerant Networks (DTNs) are a sub-class of mobile ad hoc networks (MANETs). They are mobile wireless networks that feature inherent connection disruption. In particular such networks are generally non-connected. In this paper we focus on defining a broadcast service which operate on DTNs. A number of protocols solving the problem of broadcasting across DTNs have been proposed in the past, but all of them exhibit a static behavior, i.e. they provide no control parameter. However, at the application level, flexible broadcasting schemes are desirable. In particular, it is important that the user (the source of the broadcast message) can control the way the message gets spread across the network. This paper introduces a new broadcasting protocol dedicated to DTNs, called Context-Aware Broadcasting Protocol (CABP), which adapts its greediness according to the "urgency" (priority) of the broadcast message. A formal presentation of its strategy is proposed and through preliminary experiments, the cost-effectiveness of CABP is enlightened.

1 Introduction

Mobile Ad hoc NETworks (MANETs) are wireless networks composed of nodes able to spontaneously interconnect with other nodes in their geographical neighborhood. Communication does not require any networking infrastructure since, in these networks, nodes communicate directly with each other through the radio medium. To do so, they rely on wireless networking technologies like IEEE802.11a/b/g/n (Wi-Fi) [1] or, to a lesser extent, Bluetooth [2]. When using Wi-Fi, nodes can communicate with other nodes up to a few hundred meters away, in the best case (i.e. when they use Wi-Fi in an environment free of obstacles to the propagation of radio waves).

* Partially supported by the European FET project AEOLUS.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 507–519, 2008.

MANETs are challenging networks mainly because of node mobility. Indeed, node mobility causes fluctuations of the network topology (which result, from the point of view of network nodes, in connection disruptions), as well as variations of the quality of the network links. In particular, unless specific conditions are met (even node distribution, high node density, non-standard radio signal power, etc) the network is very likely to be partitioned. When considering these challenges, mobile ad hoc networks can be referred to as Delay Tolerant Networks (DTNs), as they will in the rest of this paper.

DTNs have a variety of deployments, including Vehicular Ad hoc NETworks (VANETs) [3], sensor networks [4], military networks, etc.

This paper tackles the problem of broadcasting data across DTNs. Put in simple words, broadcasting is the process of sending one message from one node to all other nodes in the network. It has been extensively studied in the past and many broadcast protocols dedicated to mobile ad hoc networks have been proposed. Static approaches like SBA, Multipoint-Relaying [5] provide efficient solutions. Furthermore, approaches originating from distributed computing and complex systems [6] [7] were described. A recent approach, called MCB, dynamically adapts the broadcast strategy according to user-defined criteria [8]. Although it does not specifically consider preserving the network bandwidth, MCB shares some of its design objectives with the protocol presented in this paper. Unfortunately most of these protocols were designed to operate on MANETs and, because of the stronger constraints inherent to delay tolerant networking, they fail to operate in the latter context. As a consequence new protocols have to be developed for the challenging environment proposed by DTNs.

In the specific context of DTNs, the mere definition of broadcasting has to be revisited. Indeed in a DTN one cannot ensure that all nodes will be reachable. Therefore some studies tackle the broadcasting issue in a different manner. In particular, Alba and al. [9] define the message broadcasting problem as a multi-objective one consisting of:

- maximizing the number of nodes reached;
- minimizing the duration of the process;
- minimizing the bandwidth utilized.

The work presented in this article considers an extension of this definition which introduces the key notion of “message urgency”. This new parameter will directly influence the number of nodes reached, the duration of the broadcast process, and the utilization of the network bandwidth. Basically, the more urgent a message is, the greater number of nodes should be reached, the faster possible; and the lesser attention should being paid on bandwidth utilization. We call this broadcast protocol based on message urgency the “context-aware broadcast protocol” (CABP), where the urgency of the broadcast message is viewed as a context information.

The document is organized as follows. Section 2 presents the problem and the CABP protocol. Next in Section 3 the cost-effectiveness of the protocol is analyzed through simulation. Finally Section 4 concludes and presents further research directions.

2 Description of the Protocol

This section describes the Context-Aware Broadcast Protocol (CABP) by first indicating its objectives, then by detailing the strategy that it relies on, and finally by illustrating its effectiveness through simulation.

2.1 Objectives

The design objectives of CABP are threefold:

- It must operate on DTNs, given all the challenges they involve;
- it must provide the user with the ability to control its behavior, for each message processed;
- it must require little information on the network topology.

These three points are detailed in this section.

A broadcast protocol which operates on DTNs. Upon years, a fair wealth of broadcast schemes and protocols have been proposed. Most of them were designed to operate on MANETs. These protocols turn out to be inoperable on DTNs. This has motivated the development of protocols which make use of node mobility to propagate the message, such as AHBP-EX [10,11], DFCN [12] and, more generally, to epidemic broadcast schemes.

A broadcast protocol which is parameterizable. Broadcast protocols most often are targeted to providing low-level network services. In particular, broadcasting is useful in the context of routing [13] [14]. In this context, there is no need to control the behavior of broadcast protocol. This behavior is defined at the design time and cannot be altered afterwards. When looking at broadcast services from the applicative point of view, controllability turns out to be a desirable property. As an example, let us consider an industrial city surrounded by hazardous companies. In order to ensure a certain degree of safety to the population, companies have the possibility to broadcast messages across the available ad hoc networks. In the case of the formation of a toxic cloud caused by one of these companies, it is crucial that a highest-priority message is created and broadcasted across the networks, and that the propagation of this message is not slowed down by advertising messages (or more generally messages of a lower importance), or by cautious network policies whose objective is to control the usage of the bandwidth.

A broadcast protocol which require only one hop of neighborhood information. Except from Simple Flooding (a node that receives a broadcast message will forward it one single time), broadcast protocols require some form of neighborhood knowledge in order to operate. This knowledge can take the shape of *Do I have any neighbors?* or *Which are my neighbors?* or *How far is my closest neighbor?*, etc. Wu and Lou [15] have defined a classification which takes into account this amount of neighborhood knowledge that is required. They roughly define two classes: *centralized* and *localized* protocols. On the one hand, centralized protocols require global network information. Since global network information is inherently not achievable in DTNs, centralized protocols are not suitable to broadcasting in those networks. On the other hand, localized

protocols require local neighborhood information, that is information on the network topology in the first and/or second hop around the node that is executing the protocol. Protocols like AHBP-EX [16] and SBA [17] use 2-hops of neighborhood information. They exhibit the most effective strategies. However in DTNs, because of the potentially very dynamic nature of the network, 2-hops of neighborhood information may be not be achievable. In the context of DTNs, broadcasting protocols which require only 1-hop of neighborhood information are highly desirable. This is for example the case of Flooding with Self Pruning. DFCN [12] [18] and CABP are also designed in this way.

A number of broadcast protocols already meet the aforementioned design objectives; that is they operate on DTNs, they provide control on the way they behave and they require 1-hop neighborhood information. Such protocols include probabilistic schemes, distance and area-based methods [11]. Unfortunately there does not exist guidelines on how to set their parameters in order to obtain the desired effect, if possible. For example, the probabilistic scheme (nodes forwarded according to a probability defined by the user) cannot be applied in the case of the propagation of low urgency messages: experimentation showed that probabilities below 0.5 cannot be applied. As a matter of fact, metrics like network coverage or bandwidth utilization do not obey to linear functions of basic parameters such as broadcast probabilities. Recent studies [8] propose a way of parameterizing the broadcast process so as it will target to certain objectives. However, contrarily to what is presented hereinafter, these objectives do not consider the minimization of the network bandwidth.

2.2 Requirements

In order to operate, CABP requires that the nodes must:

- know the IDs of their neighbor nodes;
- locally maintain a set of node IDs associated to every message they receive;

Additionally message headers must contain:

- the ID of the node which sent the message, referred to as the *source* node (note that the source node is not the node which initiates the broadcast process, it is the one which forwards the message);
- the list of IDs of the neighbors of their source node (plus one additional word indicating the end of the neighbor list);
- one byte coding the urgency of the message;
- three bytes[1] coding the number of seconds before the message expires.

The message header should hence be structured as shown on Figure 1. Using 8 bits for encoding message urgency should provide appropriate precision in the context of this paper.

[1] Three bytes should satisfy the majority of the possible applications since it makes it possible to keeps messages almost 200 days.

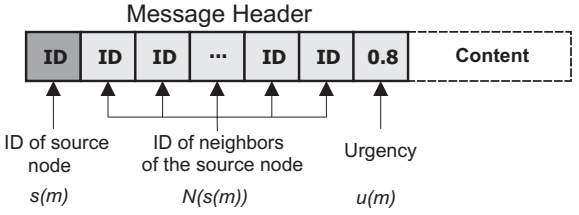


Fig. 1. The CABP messages header

2.3 Mathematical Model

In the following, we will assume that given a node n, The ID of n's neighbors is noted $N(n)$, and given a message m, the source node of m is given by $s(m)$. Additionally, the urgency of a message m is given by $u(m)$. It is defined in $[0..1]$. The greater value for $u(m)$, the greater urgency for m.

General Principle. In order to make it clear the design objectives for CABP, consider the two extreme values of importance: 0 and 1.

At the lower bound, an urgency of 0 does not imply any requirement in terms of the speed and delivery ratio of the message. In this case where the urgency is minimal, a great attention should be paid to utilizing as little resources as possible. In order to save resources, nodes forward messages with a probability that decreases when the number of their neighbors increases. In the case of 0-urgency, a node n forwards the broadcast message with a probability $P(n,m) = \frac{1}{|N(n)|^{\frac{1}{a}}}$, given that the set of known neighbors of a node n is given by $N(n)$. Hence when the neighbor density is high, individual nodes forward messages with a low probability; but the high number of nodes statistically ensures that one node will forward the message. a is used as a calibration value for the protocol. It determines how fast the forward probability decreases. For the sake of simplicity, in the following we will consider that this probability decreases in a linear fashion, that is $a = 1$.

At the upper bound, an urgency of 1 means that the message should be broadcasted at any cost, as fast as possible, and in such a manner that the delivery ratio is 100%. In that case, the resources available are utilized regardless of their utilization by other applications—which are considered of a lesser urgency. Then nodes forward the broadcast message with a probability of 1. In this extreme case, the number of neighbors is not taken into account.

Behavioral Requirements. Formally speaking, the probability $P(n,m)$ that a node n forwards a broadcast message m depends both on:

- the urgency $u(m)$ of the message m;
- the number $|N(n)|$ of neighbors of the broadcasting node n. $|N(n)|$ is defined in $[1.. + \infty]$. It is not defined below 1 because if a node n has less than one neighbor, it does not even consider forwarding messages.

$P(n,m)$ must satisfy two requirements, as defined in the following.

On the one hand, by looking at the extreme urgency values 0 and 1 as described hereinbefore, it comes that $P(m,n)$ must exhibit the following properties at the limits, as they are defined in the previous section:

$$\begin{cases} \lim\limits_{u(m)\to 0} \mathrm{P(m,\ n)} = \frac{1}{|N(n)|^{\frac{1}{a}}} \\ \lim\limits_{u(m)\to 1} \mathrm{P(m,\ n)} = 1 \end{cases}$$

On the other hand, it is desirable that $P(m,n)$ is continuous and that altering $u(m)$ impacts the behavior of the protocol in a linear manner. Indeed the behavior of the protocol is parameterized by the value of $u(m)$, whose the value is intended to be defined by a human operator. Ensuring a linear change of behavior of $P(m,n)$ when $u(m)$ varies is the best way the allow the human operator to have good control of the "urgency knob". In mathematical words, $P(m,n)$ must be a linear function of $u(m)$. That is there must exist two functions $f(n)$ and $g(n)$ so that $P(m,n) = f(n) \times u(m) + g(n)$:

Proposed Model. The most straightforward mathematical expression which meets the aforementioned requirements defines that the probability $P(n,m)$ that a node n forwards a message m is:

$$P(n,m) = \frac{1 - u(m)}{|N(n)|^{\frac{1}{a}}} + u(m)$$

Which can be put in the form $f(x) = xa + (x-1)b$, allowing $f(x)$ to morph from a to b, depending on x.

2.4 Triggers

The mathematical model described in section 2.3 is applicable in two different situations. First when a node n receives a message m from one of its neighbors, it will forward it according to a probability $P(n,m)$. Second, when a node n discovers a new neighbor, it considers forwarding every message it is currently carrying. This forward happens with the same probability $P(n,m)$.

2.5 Random Assessment Delay

Most often broadcast protocols make use of a Random Assessment Delay (commonly referred to as the RAD), which allows nodes to "wait before send". More precisely, when a node receives a broadcast message and immediately decides to forward it, it does not radio-transmit at once. Instead it will wait a random amount of time. This prevents nodes that receive simultaneously the same message from a common neighbor to forward it at the same moment. A simultaneous collective re-emission would result in a high risk of packet collision.

Broadcast protocols use a generic method for determining the assessment delay. This method consists in picking up a random number in $[0, max\_delay]$. CABP propose an extension of this strategy by benefit from nodes' neighborhood knowledge. Formally speaking, when a node n receives a message m from a source node $s(m)$, n computes an assessment delay on the basis of the neighborhood of $s(m)$. As detailed in Section

2.2, messages embed (in their header) the ordered list of neighbors' ID $N \circ s(m)$ of their source node $s(m)$. On reception of a message m, node n determines the offset $o(n, m)$ of its own ID in the list of node ID embedded in m. The assessment delay that n will wait before

$$d(n, m) = q \times o(n, m)$$

In this equation q "slices" the time, meaning that the forward of a message happens only after a delay of $n \times q$ seconds, where $n \in \mathbb{N}$. We suggest $q = 0.1s$. Note that the determination of the delay does not depend on the urgency of the message. One may think that urgent messages should be forwarded with lower delays, but doing this would increase the risk of packet collision and would finally lead to harmfully lower delivery ratio.

This technique for the determination of the assessment delay ensures a number of properties. First, if the transmission of a message lasts less than q seconds, no collision occurs. Second, the sparser is the network, the faster the message gets disseminated. In the extreme case (if no competition for the medium happens — no risk of collision exists) the message is forwarded with no delay.

2.6 Node Memory

CABP makes use of a generic technique which consists in maintaining a node-local history of the others nodes' IDs which are known to have received a given message. Basically a node remembers the nodes to which it sent the message in the past. In the same manner, it remembers the neighbors of the node which communicated him the message, since they also received it. This general technique can be applied only when 1-hop neighborhood information is available. It proves an effective way to reduce the number of transmission of broadcast messages. The technique requires that nodes individually manage an associative map

$$id_{msg} \rightarrow \{id_{node_1}, id_{node_2}, \ldots, id_{node_n}\}$$

which establishes a *one-to-n* relation from one message ID to a set of node IDs. This table is updated in the case of message emissions and receptions.

On the one hand, just before a node n emits a message m, it builds a set $N(n)$ consisting of the ID of its neighbors. These neighbors are considered to be actual recipients of the message. Then node n associates $N(n)$ to the ID of the message m, by storing the relation $m \rightarrow N(n)$ in its local associative table. Also the IDs in $N(n)$ are embedded into the message header.

On the other hand, on reception of a message m, a node n_2 obtains a list $N(s(m))$ of the neighbors of its source node $s(m)$, as well as the ID of $s(m)$. Then node n_2 stores the relation

$$m \rightarrow N \circ s(m) \cup \{s(m)\}$$

into its local associative table.

The knowledge provided by the associative table is used when a node considers forwarding a message. Before transmitting, it tests if there exists at least one of its neighbors whose the ID is *not yet* stored in the set associated to the message's ID. If one (or more) of such neighbor is found, the message is forwarded.

The lifetime of the set of IDs for a given message is the same as the lifetime of the message. As a consequence, when a message expires, all the local sets associated to it are erased from the memory of all nodes.

3 Experimentation

The behavior of CABP is investigated through simulation. This section first describes the tools that we use as well the conditions under which CABP was tested. Then preliminary results are presented.

3.1 Simulation Environment

CABP was prototyped and studied using the Madhoc wireless network simulator. Madhoc was initially targeted at the design and experimentation of broadcasting protocols. As such, it provides a framework that is suited to their development, and it comes with a set of tools that simplifies the monitoring of such highly distributed applications. In addition to that, it offers a set of mobility models allowing the simulation of a variety of environmental conditions.[2]

Our simulation campaign relied on the following parameters. The network is composed of 500 nodes evolving in a bounded squared area of $1km^2$. The nodes mobility obeys to the rules defined by the Human Mobility Model [19]. Briefly, the Human Mobility Model defines that the simulated area exhibits a set of *spots*. A spot is a circular area surrounded by a wall (walls constitute obstacles to the propagation of radio waves). Within a spot, the nodes move in random directions. When a node gets out of a spot, it chooses the closest spot that it has not yet visited. Thus every node maintains a local history of the spots they visit. Once all spots have been visited, the local history is cleared.

Although all nodes move independently from one another, the human mobility model permits the emergence of mobility patterns such as temporary group mobility, lines and clusters of nodes. The human mobility model was chosen because of its ability to reproduce such phenomenons.

The network environment we considered consists of 50 spots evenly located across the simulation area. The distance between spots is constrained so as it cannot be lower than 50m. Each spot has a radius randomly chosen between 20 and 30 meters. The graph of the initial network is represented in Figure 3. The simulation considers the broadcast of one single message, from one node to as many destination nodes as possible. Note that the initiator node is chosen so that it is in the middle of the longest path in the network graph.

3.2 Results

In order to illustrate the behavior of the CABP protocol. We will consider the following metrics:

[2] The source code of the Madhoc simulator is available at the following web address: `http://agamemnon.uni.lu/~lhogie/madhoc/`

number of nodes	500
number of spots	50
mininum dist between spots	50m
spot radius	randomly chosen in [20, 30]m
simulation area surface	$1km^2$
simulation area shape	square
message urgency	{0, 0.3, 0.6, 1}

Fig. 2. The parameters used for the experimentation campaign

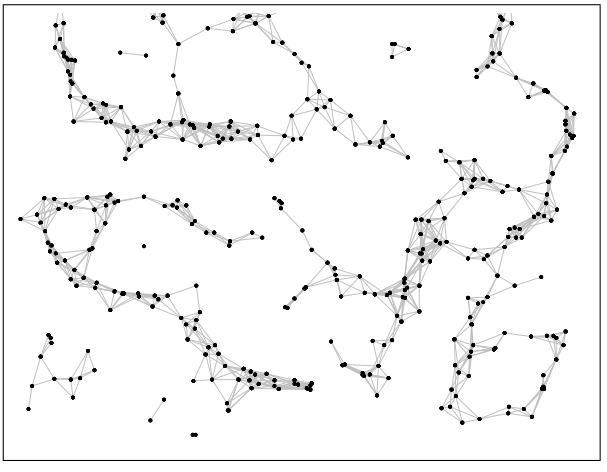


Fig. 3. The mobility model which rules the dynamics of the network defines a number of center of interests (called *spots*) where the nodes go to and stay for a while. A few parameters controlling the mobility permits to define several realistic scenarios of human mobility.

- the evolution of the coverage upon time;
- the number of message emissions upon time (this reflects the utilization of the bandwidth);
- the number of emissions carried out for reaching a given coverage;
- the evolution of the memory requirements upon time.

Bandwidth Utilization/Time. The number of emissions is an important measure because it has a direct impact on the network bandwidth which is used along the broadcasting process. The number of emissions has to be kept as low as possible, taking into account the importance of the message: the most important it is, the less care should be taken to the number of emissions. Figure 4 shows that a high urgency leads to numerous emissions, but also that it has the desirable effect to broadcast the message fast. However when reducing the importance of the message, the number of emissions dramatically lessens. This result indicates that the importance of a message should be carefully chosen. Setting a too high importance leads to a high bandwidth utilization,

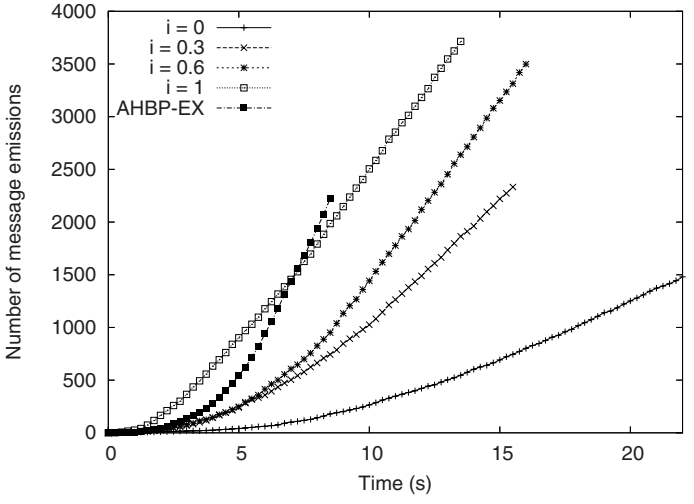


Fig. 4. If message urgency does not have such a great impact on the time required for message dissemination (makespan), it does seriously impact the bandwidth utilization. As illustrated here low-urgency messages require significantly less bandwidth to get disseminated.

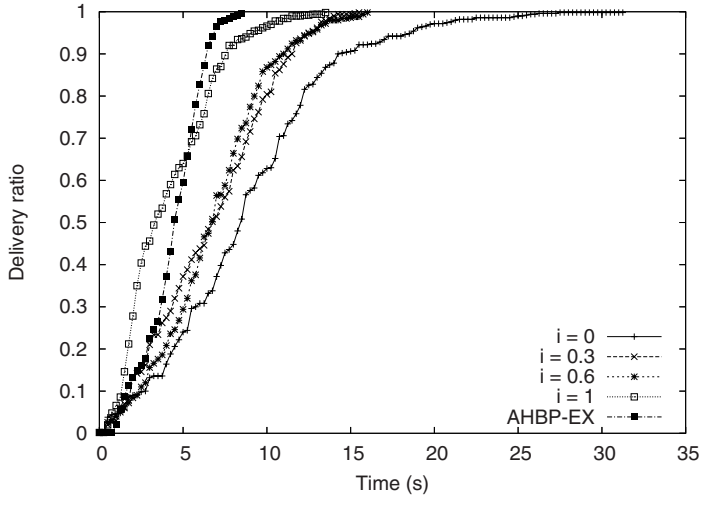


Fig. 5. The evolution of the delivery ratio depends on the message urgency. Less important messages are broadcasted using a smooth strategy whose aim is to use little network resources. A consequence is that their complete dissemination takes longer.

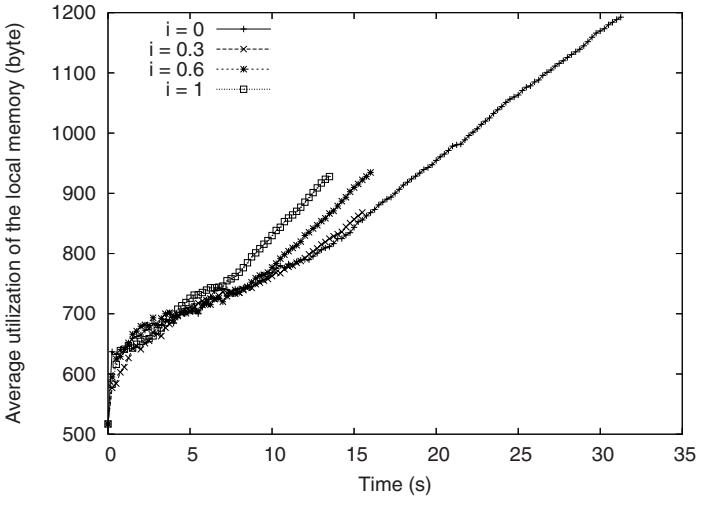


Fig. 6. The utilization of memory local to every node directly depends on the velocity of the broadcasting process. This figure illustrates the fact that the way nodes forward the broadcast messages that they hold when they meet new neighbors depends on message urgency.

while setting too low importance slightly delays the message, still ensuring a complete dissemination of the message.

Delivery Ratio/Time. Figure 5 shows the evolution of the delivery ratio upon time. The delivery ratio is the ratio of the nodes which has received the message. The simulation process is considered terminated as soon as a delivery ratio reaches a value of 1 (the message has been delivered to every node). What counts is the time required to reach a delivery ratio of 1. The more important a message is, the faster a delivery ratio of 1 should be reached. Figure 5 shows that when the importance of the message is 1, a high delivery ratio is reached fast. It also shows that this velocity of the broadcast process is not exactly proportional to the importance of the message. This attests that the probability function has room for improvement.

Note that there is no guarantee that the broadcasting process will reach every nodes. Theoretically the probability that a given node never meets another node which has received the message is not null, although insignificant. Figure 5 shows the evolution of the delivery ratio upon time. The delivery ratio is the ratio of the nodes which has received the message. The simulation process is considered terminated as soon as a delivery ratio reaches a value of 1 (the message has been delivered to every node). What counts is the time required to reach a delivery ratio of 1. The more important a message is, the faster a delivery ratio of 1 should be reached. Figure 5 shows that when the importance of the message is 1, a high delivery ratio is reached fast. It also shows that this velocity of the broadcast process is not exactly proportional to the importance of the message. This attests that the probability function has room for improvement.

Note that there is no guarantee that the broadcasting process will reach every nodes. Theoretically the probability that a given node never meets another node which has received the message is not null, although insignificant.

Local Memory Utilization. Figure 6 shows that when broadcasting in a network composed of 500 nodes moving in a 1 square kilometer area, the memory size required to store the local history for one message is significantly less than 1Kb. This value assumes that the ID of the nodes is stored on 6 bytes, as it is the case when using MAC or IPv6 addresses as nodes ID. Even if all nodes got in contact with all other nodes, they would have to store 500 IDs, which would require 3Kb of memory.

4 Conclusion and Future Works

This paper introduced the Context-Aware Broadcasting Protocol (CABP). Unlike most existing broadcast protocols, CABP is to provides a parameterizable broadcasting protocol for Mobile Ad hoc NETworks (MANETs) and Delay Tolerant Networks (DTNs).

We experimentally demonstrated that the "urgency" parameter of CABP provides the desired behavior. Indeed, the less urgent is the message, the less resources are utilized in terms of bandwidth and memory usage. On the contrary, the more urgency the message has, the quicker the broadcast process is, regardless of the resource utilized to perform it.

In addition to this, CABP proceeds regardless of the network density, which make it usable in any network condition, and in particular it can use employed in the specific context of the DTNs.

Further works include the refinement of the probabilistic model for the protocol, so that its behavior will be more linear, i.e. more controllable by the user.

References

1. ANSI/IEEE: Ansi/ieee std 802.11, 1999 edn. (r2003). wireless lan medium access control (mac) and physical layer (phy) specifications (1999)
2. The Bluetooth SIG, I.: Specification of the bluetooth system, core, vol. 1, version 1.1
3. Torrent-Moreno, M., Jiang, D., Hartenstein, H.: Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks. In: Vehicular Ad Hoc Networks, pp. 10–18 (2004)
4. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 8, 102–114 (2002)
5. Stojmenovic, I., Wu, J.: Broadcasting and activity scheduling in ad hoc networks. In: Basagni, S., Conti, M., Giordano, S., Stojmenovic, I. (eds.) Mobile Ad Hoc Networking, pp. 205–229 (2004)
6. Colagrosso, M.D.: Intelligent broadcasting inmobile ad hoc networks: three classes of adaptive protocols. EURASIP J. Wirel. Commun. Netw. 2007(1), 25–25 (2007)
7. Khelil, A.: Contact-based buffering for delay-tolerant ad hoc broadcasting. Comput. Commun. 30(16), 3144–3153 (2007)
8. Barritt, B., Malakooti, B., Guo, Z.: Intelligent multiple-criteria broadcasting in mobile ad-hoc networks. lcn 0, 761–768 (2006)

9. Alba, E., Dorronsoro, B., Luna, F., Nebro, A.J., Bouvry, P., Hogie, L.: A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan manets. Comput. Commun. 30(4), 685–697 (2007)
10. Peng, W., Lu, X.: Ahbp: An efficient broadcast protocol for mobile ad hoc networks. J. Comput. Sci. Technol. 16(2), 114–125 (2001)
11. Williams, B., Camp, T.: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), pp. 194–205 (2002)
12. Hogie, L., Bouvry, P.,, F.G.G.D.E.A.: A Bandwidth-Efficient Broadcasting Protocol for Mobile Multi-hop Ad hoc Networks. In: Demo proceeding of the 5th International Conference on Networking (ICN 2006) (October 2006). IEEE, Los Alamitos (2006)
13. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C., Jetcheva, J.: A performance comparison of multi-hop wireless ad hoc network routing protocols. In: MobiCom 1998: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp. 85–97. ACM Press, New York (1998)
14. Jones, E.P., Li, L., Ward, P.A.: Practical routing in delay-tolerant networks. In: Proceedings of the 2005 ACM SIGCOMM workshop onDelay-tolerant networking, Philadelphia, PA, USA, pp. 237–243 (August 2005)
15. Wu, J., Lou, W.: Forward-node-set-based broadcast in clustered mobile ad hoc. Wireless Communications and Mobile Computing 3, 155–173 (2003)
16. Peng, W., Lu, X.: Ahbp: An efficient broadcast protocol for mobile ad hoc networks. Journal of Computer Science and Technology 16(2), 114–125 (2001)
17. Peng, W., Lu, X.C.: On the reduction of broadcast redundancy in mobile ad hoc networks. In: MobiHoc 2000: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, pp. 129–130. IEEE Press, Los Alamitos (2000)
18. Hogie, L., Guinand, F., Bouvry, P.: A Heuristic for Efficient Broadcasting in the Metropolitan Ad Hoc Network. In: 8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES 2004), pp. 727–733 (2004)
19. Hogie, L.: Mobile Ad Hoc Networks: Modelling, Simulation and Broadcast-based Applications. PhD thesis, University of Le Havre, University of Luxembourg (April 2007)

Stability of Two-Stage Queues with Blocking

Ouiza Lekadir and Djamil Aissani

Laboratory of Modelization and Optimization of Systems
Bejaia university, Algeria, 06000
ouiza_lekadir@yahoo.fr

Abstract. Queueing networks are known to provide a useful modeling and evaluation tool in computer and telecommunications. Unfortunately, realistic features like finite capacities, retrials, priority, ... usually complicate or prohibit analytic solutions. Numerical and approximate computations as well as simplifications and performance bounds for queueing networks therefore become of practical interest. However, it is indispensable to delimit the stability domain wherever these approximations are justified.

In this paper we applied for the first time the strong stability method to analyze the stability of the tandem queues $[M/G/1 \rightarrow ./M/1/1]$. This enables us to determine the conditions for which the characteristics of the network with retrials $[M/G/1/1 \rightarrow ./M/1/1]$, can be approximated by the characteristics of the ordinary network $[M/G/1 \rightarrow ./M/1/1]$ (without retrials).

Keywords: Queueing networks, tandem queues, Stability, Retrials, Blocking, Markov chain.

1 Introduction

Tandem queueing systems arise in mathematical modeling of computer and communication networks, manufacturing lines and other systems where customers, jobs, packets, ... , are subjected to a successive processing. Tandem queues can be used for modeling real-life two-node networks as well as for validation of general networks decomposition algorithms [6]. So, tandem queueing systems have found much interest in literature. The survey of early papers on tandem queues was done in [8]. The most of these papers are devoted to the exponential queueing models. Over the last two decades, efforts of many investigators in tandem queues were directed to study a complex two tandem queues. In particular, when priority, retrials, non-exponentiality of the service, ... arises in this networks. In this cases more often than not numerical and approximate computations as well as simplifications and performance bounds for queueing networks therefore become of practical interest. However, it is indispensable to delimit the stability domain wherever these approximations are justified.

The stability analysis of queueing networks have received a great deal of attention recently. This is partly due to several examples that demonstrate that the usual conditions "traffic intensities less than one at each station" are not sufficient

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 520–529, 2008.

for stability, even under the well-known FIFO politics. Methods for establishing the stability of queueing networks have been developed by several authors, based on fluid limits [2], Lyaponov functions [5], explicit coupling (renovating event and Harris chains), monotonicity, martingales, large deviations, ...

The actual needs of practice require quantitative estimations in addition to the qualitative analysis, so in the beginning of the 1980's, a quantitative method for studying the stability of stochastic systems, called strong stability method "also called method of operators" was elaborated [1] (for full particulars on this method we suggest to see [9]). This method is applicable to all operation research models which can be governed by Markov chains.

In this article, we follow the strong stability approach to establish the stability of a two tandem queue with blocking in order to justify the approximation obtained by E. Moutzoukis and C. Langaris in [13].

The important feature and main originality of this work is that : since we establish the strong stability of the two tandem queues without intermediate space , the formulas for the characteristics of the ordinary model (without retrials) can be used to deduce the characteristics for the retrial model.

2 The Real Model

We consider a two single-server queues in tandem with blocking and retrials. Customers arrive at the first station, one a time, according to a poisson distribution with parameter λ. Each customer receives service at station 1 and then proceeds to station 2 for an additional service. There is no intermediate waiting room, so a customer whose service in the station 1 is completed can not proceed to the second station if the later is busy. Instead, the customer remains at station 1, so the last is blocked until station 2 becomes empty. The arriving customer who find the station 1 busy or blocked behave like retrial customer, he does not join a queue but he is placed instead in a hypothetical retrial queue of infinite capacity and retries for service under the constant retrial policy. According to this policy, the parameter of the exponential time of each customer in the retrial group is $\frac{\mu}{n}$, where n is the size of the retrial group. Thus, the total intensity is μ (for the different interpretation of the constant retrial policy see Farahmand [4]). If the server of station 1 is free at the time of an attempt, then the customer at the head of the retrial group receives service immediately. Otherwise, he repeats his demand later.

The service times at stations 1 and 2 are independent and arbitrarily distributed random variables with probability density functions $b_i(x)$, distribution functions $B_i(x)$ and finite mean values μ_i, for $i = 1, 2$, respectively.

2.1 The General Process

Let $X(t)$ represent the number of customers in the retrial box at time t, and for $l = 1, 2$:

$$\xi^l(t) = \begin{cases} 0 \; if \; the \; l^{th} \; server \; is \; idle \; at \; time \; t. \\ 1 \; if \; the \; l^{th} \; server \; is \; working \; at \; time \; t. \\ 2 \; if \; the \; l^{th} \; server \; is \; blocked \; at \; time \; t. \end{cases}$$

the considered model is completely describe by the regenerative process $V(t) = \left(X(t), \xi^1(t), \xi^2(t)\right)$.

2.2 The Embedded Markov Chain X_n

Denote by $d_n, n \in \mathbf{N}$, the instant of the n^{th} departure from station 1. We assume, without loss of generality, that $d_0 = 0$. If we denote $V_n = V(d_n + 0)$, then it is clear that: $V_n = \left(X(d_n + 0), \xi^1(d_n + 0), \xi^2(d_n + 0)\right) = (X_n, 0, 0)$.

So, the process V_n is a semi regenerative process with embedded Markov renewal process $(X, D) = \{X_n, d_n : n \in \mathbf{N}\}$. The last process is an irreductible and aperiodic Markov chain with the probability matrix $\mathbf{P} = \{p_{ij}\}$, where:

$$p_{ij} = \begin{cases} \int_0^\infty \frac{(\lambda t)^j}{j!} e^{-\lambda t} f_0(t)dt, & for \; i = 0, \\ \int_0^\infty \frac{(\lambda t)^{j-i}}{(j-i)!} e^{-\lambda t} f_1(t)dt + & \\ + \int_0^\infty \frac{(\lambda t)^{j-i+1}}{(j-i+1)!} e^{-\lambda t} f_2(t)dt, & for \; 1 \leq i < j+1, \\ \int_0^\infty e^{-\lambda t} f_2(t)dt, & for \; i = j+1, \\ 0, & otherwise. \end{cases}$$

where :

$$f_0(t) = \int_0^\infty \lambda e^{-\lambda w} \frac{d}{dt} \left(B_1(t)B_2(t+w)\right) dw,$$
$$f_1(t) = \int_0^\infty \lambda e^{-(\lambda+\mu)w} \frac{d}{dt} \left(B_1(t)B_2(t+w)\right) dw,$$
$$f_2(t) = \int_0^\infty \mu e^{-(\lambda+\mu)w} \frac{d}{dt} \left(B_1(t)B_2(t+w)\right) dw.$$

We define the function: $\psi_u(s) = \int_0^\infty ue^{-uw}dw \int_0^\infty e^{-sx}d_x \left(B_1(x)B_2(x+w)\right)$ and we denote :

$$\upsilon_u = \frac{-d\psi_u(s)}{ds} \mid_{s=0}; \; \rho^* = \frac{\lambda}{\lambda+\mu} + \lambda\upsilon_{\lambda+\mu}, \; \pi_k = \lim_{n\to\infty} P\left[X_n = k\right], \; k \in \mathbf{N},$$

If the intensity of the system $\rho^* < 1$, the Markov chain X_n is positive recurrent. In this case, the generating function $\Pi(z) = \sum_{n=0}^\infty \pi_n z^n$ is given by:

$$\Pi(z) = \frac{z\psi_\lambda(\lambda - \lambda z) - \left(\frac{\lambda z+\mu}{\lambda+\mu}\right) \psi_{\lambda+\mu}(\lambda - \lambda z)}{z - \left(\frac{\lambda z+\mu}{\lambda+\mu}\right) \psi_{\lambda+\mu}(\lambda - \lambda z)} \pi_0, \tag{1}$$

$$\text{where:} \qquad \pi_0 = \lim_{n\to\infty} P\left[X_n = 0\right] = \frac{1-\rho^*}{1-\rho^*+\lambda\upsilon_\lambda}. \tag{2}$$

3 The Ideal Model

We assume that the mean retrial rate in our real model tends to infinity. So, the customers in the retrial orbite try continuously to find a position for service

and they become ordinary customers. It means that if $\mu \to \infty$, our real model becomes the simple model of two queues in tandem without intermediate room, it will be referred to as an ideal model.

Now, let $\overline{X}(t)$ denote the number of customers in the first queue of the ideal model at time t and for $l = 1, 2$ we consider:

$$\overline{\xi}^{l}(t) \begin{cases} 0 \text{ if the } l^{th} \text{ server is idle at time t.} \\ 1 \text{ if the } l^{th} \text{ server is working at time t.} \\ 2 \text{ if the } l^{th} \text{ server is blocked at time t.} \end{cases}$$

Our ideal model is completely described by: $\overline{V}(t) = \left(\overline{X}(t), \overline{\xi}^{1}(t), \overline{\xi}^{2}(t)\right)$.

3.1 The Embedded Markov Chain $\overline{X}_n$

It is clear that $\overline{X}_n = (\overline{X}, D) = \{\overline{X}_n, \overline{d}_n, n \geq 0\}$ is the embedded Markov renewal process of the semiregenerative process $\left(\overline{X}(t), \overline{\xi}^{1}(t), \overline{\xi}^{2}(t)\right)$. We suppose that the intensity of the system $\rho < 1$, then $\overline{X}_n$ is an irreductible and aperiodic recurrent Markov chain with transition probability matrix $\overline{\mathbf{P}} = \left\{\overline{p}_{ij}\right\}$, where:

$$\overline{p}_{ij} = \begin{cases} \int_0^\infty \frac{(\lambda t)^j}{j!} e^{-\lambda t} f_0(t)dt, & i = 0, \\ \int_0^\infty \frac{(\lambda t)^{j-k+1}}{(j-k+1)!} e^{-\lambda t} d_t \left(B_1(t)B_2(t)\right), & 1 \leq i \leq j+1, \\ 0, & otherwise. \end{cases}$$

In this case the generating function of the v.a. $\overline{X}$ is defined as:

$\overline{\Pi}(z) = \lim_{\mu\to\infty} \Pi(z) = \frac{z\psi_\lambda(\lambda-\lambda z)-\psi(\lambda-\lambda z)}{z-\psi(\lambda-\lambda z)}\overline{\pi}_0.$
$\overline{\pi}_0 = \frac{1-\rho}{1-\rho+\lambda\upsilon_\lambda},\ \psi(s) = \int_0^\infty e^{-st} d_t \left(B_1(t)B_2(t)\right),$

$\rho = -\lambda \frac{d\psi(s)}{ds} \mid_{s=0} = \lambda \int_0^\infty t d_t \left(B_1(t)B_2(t)\right)$, We suppose that the retrial rate tends to infinity and to characterize the proximity of the ideal and real model we define the variation distance: $W = \int_0^{+\infty} \mid f_2(t) - \frac{d}{dt}\left(B_1(t)B_2(t)\right) \mid dt$.

4 The Strong Stability

This section contains preliminary results that are needed in the constructive proofs of the main theorems, given in the next sections. Let (E, ε), a measurable space, where ε is a σ-algebra denumbrably engendered. We consider a homogeneous Markov chain $Y = (Y_t, t \geq 0)$ in the space (E, ε), given by a transition kernel $\mathcal{P}(\S, \mathcal{A}), \S \in \mathcal{E}, \mathcal{A} \in \varepsilon$ and having a unique invariant probability ν.

Denote by $m\varepsilon(m\varepsilon^+)$ the space of finite (nonnegative) measures on ε and by $f\varepsilon$ $(f\varepsilon^+)$ the space of bounded measurable (nonnegative) functions on E. We

associate to every transition kernel $P(x, A)$ in the space of bounded operators, the linear mappings $\mathcal{L}_{\mathcal{P}}$ and $\mathcal{L}^*_{\mathcal{P}}$ defined by :

$$\mathcal{L}_P : \varepsilon \to m\varepsilon \qquad \mathcal{L}^*_P : f\varepsilon \to f\varepsilon$$
$$\mu \to \int_E \mu(dx)P(x, A), A \in \varepsilon \qquad f \to \int_E P(x, dy)f(y), x \in E.$$

We also associate to every function $f \in f\varepsilon$ the linear functional $f : \mu \to \mu f$ such that: $\mu f = \int_E \mu(dx)f(A);\ x \in E,\ A \in \varepsilon$.

We denote by $f \circ \mu$ the transition kernel defined as the tensorial product of the measure μ and the measurable function f having the form:

$$f(x)\mu(A);\ x \in E,\ A \in \varepsilon.$$

We consider, the Banach space $M = \{\mu \in m\varepsilon / \|\mu\| < \infty\}$, in the space $m\varepsilon$ defined by a norm $\|.\|$ compatible with the structural order in $m\varepsilon$, i.e. :

$$\begin{aligned}
\|\mu_1\| &\leq \|\mu_1 + \mu_2\|,\ for\ \mu_i \in M^+, i = 1, 2,\\
\|\mu_1\| &\leq \|\mu_1 - \mu_2\|,\ for\ \mu_i \in M^+, i = 1, 2; \mu_1 \perp \mu_2,\\
|\ \mu\ |(E) &\leq k\|\mu\|,\ for\ \mu \in M,
\end{aligned}$$

where $|\ \mu\ |$ is the variation of the measure μ, k is a finite constant and $M^+ = m\varepsilon^+ \cap M$.

The family of norms $\|\mu\|_v = \int_E v(x)\ |\ \mu\ |\ (dx), \forall \mu \in m\varepsilon$, where, v is a measurable function (not necessary finite) bounded from bellow by a positive constant, satisfy the above conditions. With this family of norms we can induce on the spaces $f\varepsilon, M$ the following norms:

$$\|P\|_v = \sup\{\|\mu P\|_v, \|\mu\|_v \leq 1 = \sup_{x \in E} \frac{1}{v(x)} \int_E |\ P(x, dy)\ | \ v(y), \tag{3}$$

$$\|f\|_v = \sup\{|\ \mu f\ |, \|\mu\|_v \leq 1\} = \sup_{x \in E} \frac{1}{v(x)} |\ f(x)\ |\ . \tag{4}$$

Definition 1. *[1] We say that the Markov chain Y, with a bounded transition kernel $\mathcal{P}$, and a unique stationary measure ν, is strongly v-stable if every stochastic kernel $\mathcal{Q}$ in the neighborhood $\{\mathcal{Q} : \|\mathcal{Q} - \mathcal{P}\|_{\sqsubseteq} < \epsilon\}$ admits a unique stationary measure $\overline{\nu}$ and :*

$$\|\nu - \overline{\nu}\|_v \to 0 \quad when \quad \|\mathcal{Q} - \mathcal{P}\|_v \to 0$$

Theorem 1. *[1] The Harris recurrent Markov chain Y with a bounded transition kernel $\mathcal{P}$, and a unique stationary measure ν, is strongly v-stable, if the following conditions holds:*

1. $\exists\ \alpha \in\ M^+, \exists\ h \in f\varepsilon^+ / \pi h > 0, \alpha\ \mathbf{I} = 1, \alpha h > 0,$
2. $T = \mathcal{P} - h \circ \alpha \geq 0,$
3. $\exists\ \gamma < 1 /\ Tv(x) \leq \gamma v(x), \forall x \in E,$

where $\mathbf{I}$ is the function identically equal to 1.

Theorem 2. *Under the conditions of the theorem (1) and if Δ (the deviation of the operator transition $\mathcal{P}$) verifying the condition $\|\Delta\|_v < \frac{1-\gamma}{C}$, we have:* $\|\nu - \overline{\nu}\|_v \leq \|\Delta\|_v \|\nu\|_v C\big(1-\gamma - C\|\Delta\|_v\big)^{-1},\ C = 1 + \|\mathbf{I}\|_v\|\nu\|_v$.

5 Stability of the Ideal Model

We define on $E = \mathbf{N}$ the σ-algebra ε engendered by the set of all singletons $\{j\}, j \in \mathbf{N}$. We consider the function $v(k) = \beta^k, \beta > 1$ and we define the norm: $\|\mu\|_v = \sum_{j\in\mathbf{N}} v(j) \mid \mu \mid (\{j\}), \forall \mu \in m\varepsilon$. We also consider the measure $\alpha(\{j\}) = \alpha_j = \overline{p}_{0j}$, and the measurable function $h(i) = \begin{cases} 1 & \text{if } i = 0, \\ 0 & \text{otherwise.} \end{cases}$.
Using the assumptions we obtain the following lemmas

Lemma 1. *Let $\overline{\pi}$ the stationary distribution of the Markov chain $\overline{X}_n$, then:*

$$\alpha\mathbf{I} = 1,\ \alpha h > 0 \text{ and } \alpha\overline{\pi} > 0.$$

Proof. It is easy to show that :
- $\alpha\mathbf{I} = \sum_{j=0}^{\infty} \alpha(\{j\}) = \sum_0^{\infty} p_{oj} = 1.$
- $\alpha h = \sum_{j=0}^{\infty} \alpha(\{j\})h(j) = p_{00} > 0.$
- $\overline{\pi}h = \sum_{i=0}^{\infty} \overline{\pi}_i h(i) = \overline{\pi}_0 = \frac{1-\rho}{1-\rho+\lambda v_\lambda} > 0.$

Lemma 2. *Suppose that the following conditions holds:*

1. $\lambda \int_0^{+\infty} ud\,(B_1(u)B_2(u)) < 1$, *(Geometric ergodicity condition).*
2. $\exists a > 0 / \int_0^{\infty} e^{au} d\,(B_1(u)B_2(u)) < +\infty$, *(Cramer condition).*
3. $\int_0^{+\infty} t \mid f_2(t) - \frac{d}{dt}\,(B_1(t)B_2(t)) \mid dt < \frac{W}{\lambda}$.

then, $\exists \beta > 1$ such that:

- $\frac{\psi(\lambda-\lambda\beta)}{\beta} < 1.$
- $\int_0^{+\infty} e^{(\lambda\beta-\lambda)t} \mid f_2(t) - \frac{d}{dt}\,(B_1(t)B_2(t)) \mid dt < \beta W.$

Where W is given by the formula (3.1).

Proof.
• We consider the function: $K(\beta) = \psi(\lambda - \lambda\beta)$. K is continuous differentiable in $[1, a]$, so: $K'(\beta) = \lambda \int_0^{+\infty} te^{(\lambda\beta-\lambda)t} d_t\,(B_1(t)B_2(t))\,,$
$K''(\beta) = \lambda^2 \int_0^{+\infty} t^2 e^{(\lambda\beta-\lambda)t} d_t\,(B_1(t)B_2(t))\,,$
then, K is a strictly convex function in $[1, a]$. We define the function :

$$L(\beta) = \frac{K(\beta)}{\beta} = \frac{1}{\beta}\int_0^{+\infty} e^{(\lambda\beta-\lambda)t} d_t\,(B_1(t)B_2(t))\,.$$

For $\beta = 1$, $L(1) = \psi(0) = 1$ and for $1 < \beta < a$, $L'(\beta) = \frac{\lambda\beta\psi(\lambda-\lambda\beta)-\psi(\lambda-\lambda\beta)}{\beta^2}$,
From the first assumption we have:

$$L'(1) = \lambda\psi'(0) - \psi(0) = \lambda \int_0^{+\infty} ud\,(B_1(u)B_2(u)) - \psi(0) < 0.$$

So in the vicinity of 1, L is decreasing. Then, $\exists \beta > 1$ such that:
$L(\beta) < L(1) \Rightarrow L(\beta) < 1$, so : $\exists \beta > 1 :\ \frac{\psi(\lambda-\lambda\beta)}{\beta} < 1$. Moreover, let's consider:

$$\beta_0 = \sup\{\beta : \psi(\lambda\beta - \lambda) < 1\}, 1 < \beta_0 < +\infty. \tag{5}$$

The convexity of the function $K(\beta)$ imply that :

$$\psi(\lambda\beta-\lambda)<\beta, \forall\beta\in\,]1,\beta_0\,]\Rightarrow\psi(\lambda\beta-\lambda)<\beta_0,$$

• We put : $\varphi(\lambda\beta-\lambda)=\int_0^{+\infty}e^{(\lambda\beta-\lambda)t}\mid f_2(t)-\frac{d}{dt}\left(B_1(t)B_2(t)\right)\mid dt$ and we consider the function $\Omega(\beta)=\varphi(\lambda-\lambda\beta)$.

For $\beta=1:\Omega(1)=\varphi(0)=\int_0^{+\infty}\mid f_2(t)-\frac{d}{dt}\left(B_1(t)B_2(t)\right)\mid dt=W$.

For $1<\beta<a$, Ω is continuous and differentiable, so: $\Omega'(\beta)=\frac{\beta\varphi'(\lambda\beta-\lambda)-\varphi(\lambda\beta-\lambda}{\beta^2}$. The functions φ and φ' are continuous, then:

$$\lim_{\beta\to1^+}\Omega'(\beta)=\lim_{\beta\to1^+}\left[\lambda\varphi'(\lambda\beta-\lambda)-\varphi(\lambda-\lambda\beta)\right]=\lambda\varphi'(0^+)-\varphi(0^+)\tag{6}$$

$\varphi'(0^+)=\int_0^{+\infty}t\mid f_2(t)-\frac{d}{dt}\left(B_1(t)B_2(t)\right)\mid dt$ and $\varphi(0^+)=W$. From the third assumption we have :

$$\lambda\int_0^{+\infty}t\mid f_2(t)-\frac{d}{dt}\left(B_1(t)B_2(t)\right)\mid dt<W,$$

so $\Omega'(1^+)<0$ then $\varphi(\beta)-\varphi(1)<0$ in the vicinity of 1. It means that $\exists\beta>1$ such that $\Omega(\beta)=\frac{\varphi(\lambda\beta-\lambda)}{\beta}<W$.

Lemma 3. *The operator $\overline{T}=\overline{\mathbf{P}}-h\circ\alpha$ is nonnegative and $\exists\gamma<1$, such that $\overline{T}v(k)\le\gamma v(k)$ for all $k\in\mathbf{N}$.*

Proof. We have $\overline{T}(i,\{j\})=\overline{T}_{ij}=\overline{p}_{ij}-h(i)\alpha(\{j\})$, so :

$$\overline{T}_{ij}=\begin{cases}0, & \text{if } i=0,\\ \overline{p}_{ij}\ge0, & \text{if } i\ge1.\end{cases}\Rightarrow\overline{T}\text{ is non negative.}$$

Let's compute $\overline{T}v(k)$:
If $k=0$, we have $\overline{T}v(0)=0$. If $k\neq0$, we have :

$$\begin{aligned}\overline{T}v(k)&=\sum_{j\ge0}\beta^jT_{kj},1\le k\le j+1,\\&=\beta^{k-1}\int_0^{+\infty}e^{(\lambda\beta-\lambda)t}d_t\left(B_1(t)B_2(t)\right)=\beta^{k-1}\psi(\lambda-\lambda\beta).\end{aligned}$$

We consider that $\gamma=\frac{\psi(\lambda-\lambda\beta)}{\beta}$. From the lemma (2), $\exists\beta\in]1,\beta_0]$ such that $\gamma<1$. So, there exists β with $1<\beta\le\beta_0$ such that:

$$\overline{T}v(k)\le\gamma v(k),\forall k\in\mathbf{N},\gamma=\frac{\psi(\lambda-\lambda\beta)}{\beta}<1.\tag{7}$$

Lemma 4. *The norm of the transition kernel of the chain $\overline{X}_n$ is bounded.*

Proof. We have: $\|\overline{\mathbf{P}}\|=\|\overline{T}+h\circ\alpha\|_v\le\|\overline{T}\|_v+\|h\|_v\|\alpha\|_v$.

$$\bullet\|\overline{T}\|_v=\sup_{k\ge0}\frac{1}{v(k)}\sum_{j\ge0}v(j)\mid\overline{T}_{kj}\mid=\gamma<1,$$

$$\bullet\|h\|_v=\sup_{k\ge0}1/v(k)=\frac{1}{\beta^k}=1,$$

$$\bullet \|\alpha\|_v = \sum_{j\geq 0} \beta^j \int_0^\infty \frac{(\lambda t)^j}{j!} e^{-\lambda t} f_0(t)dt = \int_0^\infty e^{(\beta\lambda-\lambda)t} dF(t) < \infty,$$

$$\text{with } F(t) = \int_0^\infty \lambda e^{-\lambda w} \left(B_1(t)B_2(t+w)\right) dw.$$

So: $\|\overline{\mathbf{P}}\|_v \leq 1 + \beta_0 < \infty$.

Theorem 3. *In the two tandem queues with blocking, the Markov chain $\overline{X}_n$ representing the number of customers in the first station at the instant of the n^{th} departure from the first station, is strongly v-stable with respect to the function $v(k) = \beta^k$ for all $1 < \beta \leq \beta_0$. Where β_0 is given by the formula (5).*

Proof. The proof arises from the theorem 1. Indeed, all necessary conditions to establish the v-strong stability required in the theorem 1 are satisfied and are given by the above lemmas (1), (3), (4).

6 Deviation of the Transition Operator

Lemma 5. *Let $\mathbf{P}$ (resp. $\overline{\mathbf{P}}$) be the transition operator associate to the Markov chain X_n (resp. $\overline{X}_n$). Then: $\left\|P - \overline{P}\right\|_v \leq W + \int_0^\infty e^{(\beta\lambda-\lambda)t} f_1(t)dt$.*

Proof. We have :

$$\left\|P - \overline{P}\right\|_v = \sup_{k\geq 0} \frac{1}{v(k)} \sum_{j\geq 0} v(j) \mid p_{kj} - \overline{p}_{kj} \mid, \sup_{k\geq 0} \frac{1}{\beta^k} \sum_{j\geq 0} \beta^j \mid p_{kj} - \overline{p}_{kj} \mid,$$

$$= \sup \left(0, \quad \sup_{k>0} \frac{1}{\beta^k} \sum_{j\geq 0} \beta^j \mid p_{kj} - \overline{p}_{kj} \mid\right),$$

We put $Q(k) = \sum_{j\geq 0} \beta^j \left|p_{kj} - \overline{p}_{kj}\right|$. If $k \neq 0$ we have $1 \leq k \leq j+1$, so :

$$Q(k) = \sum_{j\geq 0} \beta^j \mid p_{kj} - \overline{p}_{kj} \mid = \sum_{j\geq k-1} \beta^j \mid p_{kj} - \overline{p}_{kj} \mid,$$

$$= \beta^{k-1} \mid p_{kk-1} - \overline{p}_{kk-1} \mid + \sum_{j\geq k} \beta^j \mid p_{kj} - \overline{p}_{kj} \mid,$$

$$\leq \beta^k \left[\frac{\int_0^\infty e^{(\beta\lambda-\lambda)t} \left|f_2(t) - \frac{d}{dt}\left(B_1(t)B_2(t)\right)\right| dt}{\beta} + \int_0^\infty e^{(\beta\lambda-\lambda)t} f_1(t)dt\right].$$

Using the lemma (2), we obtain:
$\left\|P - \overline{P}\right\|_v \leq W + \int_0^\infty e^{(\beta\lambda-\lambda)t} f_1(t)dt$, with: $\lim_{\mu\to\infty} \int_0^\infty e^{(\beta\lambda-\lambda)t} f_1(t)dt = 0$.

7 Deviation of the Stationary Distribution

Theorem 4. *Let's π (resp. $\overline{\pi}$) the stationary distribution of the real model, $[M/G/1/1 \to ./G/1/1]$ with retrials, (resp. the ideal model $[M/G/1 \to ./G/1/1]$).*

For $1 < \beta < \beta_0$ *and* $\|\Delta\|_v < \frac{1-\gamma}{1+c_0}$, *we have:*
$\|\pi - \overline{\pi}\|_v \leq c_0(1+c_0)\|\Delta\|_v\big(1-\gamma-(1+c_0)\|\Delta\|_v\big)^{-1}$, *where* $c_0 = \frac{\psi_\lambda(\lambda\beta-\lambda)-\gamma}{1-\gamma}$.

Proof. From the theorem ([2]) we have :

$$\|\pi - \overline{\pi}\|_v \leq \|\Delta\|_v\|\overline{\pi}\|_v C\big(1-\gamma-C\|\Delta\|_v\big)^{-1}.$$

Or we have : $\|\overline{\pi}\|_v = \sum_{j\geq 0} v(j)\overline{\pi} = \overline{\Pi}(\beta)$
$= \frac{\beta\psi_\lambda(\lambda\beta-\lambda)-\psi(\lambda\beta-\beta)}{\beta-\psi(\lambda\beta-\lambda)} = \frac{\psi_\lambda(\lambda\beta-\lambda)-\gamma}{1-\gamma} = c_0.$
and $\|\mathbf{I}\|_v = \sup_{k\geq 0}\frac{1}{\beta^k} = 1$ So:
$C = 1 + \|\mathbf{I}\|_v\|\overline{\pi}\|_v = 1 + \frac{\psi_\lambda(\lambda-\lambda\beta)-\gamma}{1-\gamma} = 1 + c_0$. Finally, we obtain :

$$\|\pi - \overline{\pi}\|_v \leq c_0(1+c_0)\|\Delta\|_v\big(1-\gamma-(1+c_0)\|\Delta\|_v\big)^{-1}.$$

8 Conclusion

This work is a first attempt to prove the applicability of the strong stability method to a queueing networks. We have obtained the conditions under which the characteristics of the tandem queues $[M/G/1/1 \rightarrow ./M/1/1]$ with retrials can be approximated by those of the ordinary network $[M/G/1 \rightarrow ./M/1/1]$ (without retrials). This allow us to justify the approximation established by E. Moutzoukis and C. Langaris in [13].

In term of prospect, we propose to work out an algorithm which checks the conditions of approximation of these two tandem queues and determine with precision the values for which the approximation is possible. It will also determine the error on the stationary distribution which had with the approximation.

References

1. Aïssani, D., Kartashov, N.V.: Ergodicity and stability of Markov chains with respect to the topology in the space of transition kernels. Doklady Akademii Nauk Ukrainskoi SSR seriya A 11, 3–5 (1983)
2. Dai, J.G.: On positive Harris recurrence of multiclass queueing networks: A unified approch via limite fluid limit models. Annals of Applied Probability 5(1), 49–77 (1993)
3. Dallery, Y., Gershwin, B.: Manufacturing flow line systems: a review of models and analytical results. Queueing systems 12, 3–94 (1992)
4. Faramand, F.: Single line queue with repeated demands. Queueing Systems 6, 223–228 (1990)
5. Fayolle, G., Malyshev, V.A., Menshikov, M.V., Sidorenko, A.F.: Lyaponov functions for Jackson networks. Rapport de recherche 1380, INRIA, Domaine de Voluceau, LeChenay (1991)
6. Ferng, H.W., Chang, J.F.: Connection-wise end-to-end performance analysis of queueing networks with MMPP inputs. Performance Evaluation 43, 362–397 (2001)
7. Foster, F.G., Perros, H.G.: On the blocking process in queue networks. Eur. J. Oper. Res. 5, 276–283 (1980)

8. Gnedenko, B.W., Konig, D.: Handbuch der Bedienungstheorie. Akademie Verlag, Berlin (1983)
9. Kartashov, N.V.: Strong stable Markov chains. VSP. Utrecht. TBIMC. Scientific Publishers (1996)
10. Karvatsos, D., Xenios, N.: MEM for arbitrary queueing networks with multiple general servers and repetative service blocking. Performance Evaluation 10, 169–195 (1989)
11. Kerbache, L., Gregor, S.J.M.: The generalized expansion method for open finite queueing networks. Eur. J. Oper. Res. 32, 448–461 (1987)
12. Li, Y., Cai, X., Tu, F., Shao, X., Che, M.: Optimisation of tandem queue systems with finite buffers. Computers and Operations Research 31, 963–984 (2004)
13. Moutzoukis, E., Langaris, C.: Two queues in tandem with retrial customers. Probability in the Engineering and Informational Sciences 15, 311–325 (2001)
14. Pellaumail, J., Boyer, P.: Deux files d'attente á capacité limitée en tandem. Tech. Rept. 147: CNRS/INRIA. Rennes (1981)
15. Hillier, F., So, K.: On the optimal design of tandem queueing systems with finite buffers. Queueing systems theory application 21, 245–266 (1995)
16. Huntz, G.C.: Sequential arrays of waiting lines. Operations Research 4, 674–683 (1956)
17. Avi-Itzhak, B., Yadin, M.: A sequence of two servers with no intermediate queue. Management Science 11, 553–564 (1965)
18. Suzuki, T.: On a tandem queue with blocking. Journal of the Operations Research Society of Japon 6, 137–157 (1964)

A Competitive Neural Network for Intrusion Detection Systems

Esteban José Palomo, Enrique Domínguez, Rafael Marcos Luque, and José Muñoz

Department of Computer Science
E.T.S.I.Informatica, University of Malaga
Campus Teatinos s/n, 29071 – Malaga, Spain
{ejpalomo,enriqued,rmluque,munozp}@lcc.uma.es

Abstract. Detecting network intrusions is becoming crucial in computer networks. In this paper, an Intrusion Detection System based on a competitive learning neural network is presented. Most of the related works use the self-organizing map (SOM) to implement an IDS. However, the competitive neural network has less complexity and it is faster than the SOM, achieving similar results. In order to improve these results, we have used a repulsion method among neurons to avoid overlapping. Moreover, we have taken into account the presence of quantitative data in the input data, and they have been pre-processed appropriately to be supplied to the neural network. Therefore, the current metric based on Euclidean distance to compare two vectors can be used. The experimental results were obtained by applying the KDD Cup 1999 benchmark data set, which contains a great variety of simulated networks attacks. Comparison with other related works is provided.

Keywords: Competitive learning, network security, intrusion detection system, data mining.

1 Introduction

As communications among computer networks grow, computer crimes increase and network security becomes more difficult. Intrusion detection systems (IDS) monitor network traffic to detect intrusions or attacks. The two main detection methods of an IDS are misuse detection and anomaly detection. The misuse detection method detects attacks storing the signatures of previously known attacks. This method fails detecting new attacks and to include them, the signature database has to be manually updated. Anomaly detection is another approach, where a normal profile is established. Then, deviants from normal profile are detected as attacks. Some anomaly detection systems using data mining techniques such as clustering, support vector machines (SVM) and neural network systems have been proposed [1,2,3]. The artificial neural networks provide many advantages in the detection of network intrusions [4]. Neural network models have usually been applied for misuse detection and anomaly detection.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 530–537, 2008.

The self-organizing map (SOM) has been used in anomaly detection since it constitutes an excellent tool for data mining, knowledge discovery and preserves the topology of the input data [5]. However, the SOM has an important drawback: the computation time. Indeed, the number of neurons decreases the network's performance. In this paper, we have used a competitive learning neural network to build an IDS, since this kind of neural network has less complexity than the SOM and the results achieved are similar to that achieved with SOMs. Moreover, we have taken into account the presence of symbolic data among the input data, which have to be processed before being supplied as input data to the neural network.

The IDS based on a competitive learning neural network (CLNN) was trained with the KDD Cup 1999 benchmark data [6,7]. This data set has served as one of the most reliable benchmark data set that has been used for most of the research work on intrusion detection algorithms [8]. It consists of connection records preprocessed from network traffic, where several attacks were simulated.

The rest of this paper is organized as follows. In the next Section, we provide a description of the proposed approach. Section 3 presents some experimental results obtained after testing our IDS with the KDD Cup 1999 benchmark. Then, these results are compared to other related works. Section 4 concludes the current paper.

2 Competitive Model

The learning process of a competitive neural network can be supervised or unsupervised. In the supervised learning process, the input data must be labeled, whereas the unsupervised learning process can use unlabelled input data. Unsupervised learning algorithms have been extensively used to face data clustering over several decades [9]. We have built our IDS facing the problem of distinguishing attacks from normal records as a clustering problem, where we have as much clusters as different connection types we want to detect.

The proposed competitive learning neural network (CLNN) consists of a single layer of n neurons, where each neuron has assigned a vector of m features, called weight vector. The weight vectors are initialized with randomly selected input patterns. The output neurons compete among themselves when an input pattern is presented. The neuron with the smallest Euclidean distance between its weight vector w_i and the current input pattern x becomes the winner. The winner's weight vector is updated in order to approach the current input pattern, following a learning rate α decreasing with time. The weight vector of the winner is updated using the competitive learning rule:

$$w_r(t+1) = w_r(t) + \alpha(t)[x(t) - w_r(t)] \tag{1}$$

where α is the learning rate, w_r is the weight vector of the winning neuron r, x is the input data, and t is the current time-step. The winner neuron represents the closest cluster to the current input pattern and moves its weight vector to the

input pattern. Therefore, after training each output neuron represents a cluster of the input data set, and their weight vectors are in the centre of each cluster.

One shortcoming that arises is that different clusters can overlap, involving bad clustering of the input data. For that reason, we have used a repulsion rule to the winner on the rest of neurons. Thus, once the winner is updated, the rest of neurons are moved further away from the winner as long as the module of the difference between their weight vector w_i and the winner's weight vector w_r is less than a certain threshold Δ, as shown in (2). The neurons that satisfy the previous condition, are moved further away from the winner neuron guided by a repulsion rate β, as given in (3). This way, although it does not guarantee optimal separation among clusters, it does avoid being clusters less than a specified distance Δ.

$$\|w_r - w_i\| < \Delta \tag{2}$$

$$w_i(t+1) = w_i(t) - \beta(t)[w_r(t) - w_i(t)] \tag{3}$$

In order to compare the weight vectors with the input data, the Euclidean distance has been used. Obviously, this distance measure can just be applied to numerical data. However, in many applications symbolic data can exist in addition to numerical data. That is the case of the Intrusion Detection Systems, where we can find the values TCP, UDP i.e., as symbolic data from the 'protocol_type' feature. In such conditions, many related works have mapped these qualitative data into quantitative values [10,8,11]. Although using this mapping we can apply the Euclidean distance, symbolic values are also assigned a distance among them, when symbolic values must just indicate whether they are present or not and have no any distance associated.

This problem has been solved replacing the symbolic feature with as many new binary features as possible values of that feature are. For example, if a symbolic feature can have six different values, the feature will be replaced with six new binary features, so that each feature represents a possible value of the symbolic feature. Thus, in order to indicate a value of the symbolic feature, its corresponding binary feature will be 1 and the rest of the new features will be 0. This way, we can use Euclidean distance whereas it is shown whether the symbolic values are present or not. The pseudo code of the algorithm is shown in the figure 1.

3 Experimental Results

The neural network was trained and tested with the KDD Cup 1999 benchmark data set created by MIT Lincoln Laboratory and available on the University of California, Irvine site [12]. This data set was used for the Third International Knowledge Discovery and Data Mining Tools Competition to detect simulated attacks in a network environment. For training, we have used the 10% KDD Cup 1999 training data set, which contains 494021 connection records, each with 41

Input: $X = x_1, x_2, ..., x_m$
Output: $W = w_1, w_2, ..., w_n$

BEGIN
Randomly initializate the weight vectors $w_i, i = 1, 2, ..., n$.
Normalize the weight vectors $w_i \in [0, 1]$.
Initialize the learning rate α, $\alpha < 1$.
for $x_j \in X$ **do**
for $w_i \in W$ **do**
/* Compute distances */
$d(x_j, w_i) = \|x_j - w_i\|$
end for
/* Compute the winner index */
$r = arg\min_i d(x_j, w_i)$
/* Update the weight vector of the winner */
$w_r = w_r + \alpha(x_j - w_r)$
for $w_i \in W$ **do**
if $\|w_r - w_i\| < \Delta$ **then**
/* Update the weight vector of the neuron */
$w_i = w_i - \beta(w_r - w_i)$
end if
end for
end for
END

Fig. 1. Pseudocode of the proposed competitive model

features. It contains 22 different attack types and normal records. The 22 attack types fall into four main categories [13]:

- **Denial of Service (DoS):** an attempt to make a computer resource unavailable to prevent legitimate users from using that resource.
- **Probe:** the location of weak points by mapping the machines and services that are available on a network.
- **Remote-to-Local (R2L):** occurs when an unauthorized attacker from an outside system exploits some vulnerability to gain local access as a user of that machine.
- **User-to-Root (U2R):** occurs when an attacker with an user account on the system is able to exploit some vulnerability to gain root access to the system.

All the 41 features are numerical except three of them which are symbolic: protocol type (i.e. TCP, UDP, ...), service (i.e. telnet, ftp, ...) and status of the

connection flag. These features have to be mapped to numerical values in order to compare two vectors with the Euclidean distance. However, as we mentioned in Section 2, it makes no sense to map qualitative values into quantitative values since it assigns an order among symbolic values of a feature. For that reason, each symbolic feature has been replaced for new binary features, according to the number of possible values that each feature can has. In the training data set, the protocol type feature has 3 different values, the service feature 66 and the status of the connection flag feature 11. Thus, we increase the number of features from 41 to 118. Initially, a new feature has assigned the value 1 if the replaced symbolic feature had assigned the value that represents that new feature or 0 otherwise. Thus, each symbolic value is mapped into a quantitative value without assigning an order among them.

In order to train our competitive neural network (CLNN), we have selected two different subsets, S1 and S2, from the 494,021 connection records in the training data set. Both subsets contain the 22 attack types and normal records with a total of 100,000 and 169,000 connection records, respectively. After training, the proposed neural network was tested with the entire 10% KDD Cup 99 testing data set, which is composed of 311,029 connection records. In this test data set, we find 15 new attack types which are not found in the training data set and, for that reason we do not know their attack category. The distribution of the different data subsets is shown in Table 1, where the new attack types existing in the test data set are categorized as 'Unknown'.

During the training, each connection record was randomly selected from their corresponding data subset. We used 7 neurons to detect each attack category and normal records, and we establish the threshold for the repulsion condition to $\Delta = 1$. The training results of both subsets, simulated with the same training data sets, are given in Table 2. Here, the detected rate is the ratio of the attacks that were detected, the false positive rate is the ratio of the normal connection records that were detected as attacks, and the identified rate is the ratio of the connection records that were identified as their correct category, taking into account the four attack categories and the category of normal connection records. The computation time in the training phase was 31.55 and 108.35 seconds for S1 and S2, respectively, where 2 epochs were used. After training, the testing was done with the 10% testing data set. These testing results are shown in Table 3.

On examining the results in Table 3, we achieved 99.99% detection rate for both subsets, and false positive rates between 4.25% and 3.98%, respectively. In order to build an IDS, an improved competitive learning network (ICLN) was used in [10]. Their best result was 97.89% detection rate (we do not now the false positive rate), but using between 9 and 20 neurons, whereas we have used 7 neurons. Also, they just used 7 attack types instead of the 22 attack types. Most of the related works have used the self-organizing map (SOM) in network security. In [8], a hierarchical Kohonen Map (K-Map) was proposed as an IDS. It consists of three layers, where each layer is a single K-Map or SOM. They achieved 99.63% detection rate and a false positive rate of 0.63% as best results,

Table 1. Data distribution of different data subsets

Connection Category	10% Training	S1	S2	10% Test
Normal	97278	30416	53416	60593
DoS	391458	64299	110299	223298
Probe	4107	4107	4107	2377
R2L	1126	1126	1126	5993
U2R	52	52	52	39
Unknown	0	0	0	18729

Table 2. Training results for S1 and S2

Training Set	Detected(%)	False Positive(%)	Identified(%)
S1	99.99	0.66	94.76
S2	99.99	0.81	95.31

Table 3. Testing results for the proposed neural network

Training Set	Testing Set	Detected(%)	False Positive(%)	Identified(%)
S1	10% KDD Test	99.99	4.25	90.18
S2	10% KDD Test	99.99	3.98	90.24

Table 4. Comparison results for differents IDS implementations

	Detected(%)	False Positive(%)	Neurons
CLNN	99.99	3.98	7
ICLN	97.89	-	15-20
K-Map	99.63	0.34	144
SOM	97.31	0.042	400
SOM (DoS)	99.81	0.1	28800-36000

but having several limitations. They used a training set of 169,000 connection records and 22 attack types as we used. However, it was tested with just three attack types, using a pre-specified combination of 20 features, concretely, 4 for the first layer, 7 for the second and 9 for the third. Moreover, these three connected SOMs were established in advance using 48 neurons in each level. In addition to the limitations and the complexity of the net because of the number of neurons used, it involves a pre-processing and a study of the problem domain. An ensemble of self-organizing maps was also used in [13], in order to implement an IDS. From this work, the SOM trained on all features with the best results was chosen for comparing purposes. The SOM achieved a detection rate of 97.31% and a false positive rate of 0.042% by using 400 neurons, whereas we used just 7 neurons improving the performance of the neural network. Another SOM was

used in [14], providing detection rates that ranges between 98.3% and 99.81% and false positive rates between 2.9% and 0.1%. However, it is just limited to DoS attacks and the number of neurons ranges between 160x180 and 180x200 neurons. Furthermore, their best result (9.81% detection rate and 0.1% false positive rate) was achieved by using just the one attack type (smurf), both in the training and testing data set. Table 4 reports the performances and the number of neurons used of the different mentioned IDS.

4 Conclusions

An instrusion detection system based on a competitive learning neural network has been proposed in this paper. The simplicity of its architecture increases the speed of the performance of the neural network. A repulsion mechanism based on threshold Δ is provided to avoid overlapping clusters. Moreover, we have taken into account the presence of qualitative data in the input data set. Unlike other related works that map these qualitative data into quantitative data, we have extended the dimensionality of the input data to include one feature for each possible symbolic value. Thus, symbolic values can be used with the current metric based on Euclidean distance.

In order to train and test our IDS, we have used the KDD Cup 1999 benchmark data set. This data set contains both, qualitative and quantitative values, and has been used for most of the research work on intrusion detection systems. We have used the 10% KDD Cup 1999 training data set and the 10% KDD Cup 1999 testing data for training and testing, respectively. From the training data set, we selected two subsets with 100,000 and 169,000 connection records and the 22 attack types existing in the training data set. The trained neural networks were tested with the entire testing data set, which is composed of 311,029 connection records and contains 15 new attack types. We achieved detection rates of 99.99% and false positive rates between 3.98% and 4.63% with just 7 neurons.

Acknowledgements

This work is partially supported by Spanish Ministry of Education and Science under contract TIN-07362.

References

1. Lee, W., Stolfo, S., Chan, P., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J.: Real time data mining-based intrusion detection. In: DARPA Information Survivability Conference and Exposition II, vol. 1, pp. 89–100 (2001)
2. Maxion, R., Tan, K.: Anomaly detection in embedded systems. IEEE Transactions on Computers 51(2), 108–120 (2002)
3. Tan, K., Maxion, R.: Determining the operational limits of an anomaly-based intrusion detector. IEEE Journal on Selected Areas in Communications 21(1), 96–110 (2003)

4. Cannady, J.: Artificial neural networks for misuse detection. In: Proceedings of the 1998 National Information Systems Security Conference (NISSC 1998), Arlington, VA, October 5-8, 1998, pp. 443–456 (1998)
5. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological cybernetics 43(1), 59–69 (1982)
6. Lee, W., Stolfo, S., Mok, K.: A data mining framework for building intrusion detection models. In: IEEE Symposium on Security and Privacy, pp. 120–132 (1999)
7. Stolfo, S., Fan, W., Lee, W., Prodromidis, A., Chan, P.: Cost-based modeling for fraud and intrusion detection: results from the jam project. In: DARPA Information Survivability Conference and Exposition, 2000. DISCEX 2000. Proceedings, vol. 2, pp. 130–144 (2000)
8. Sarasamma, S., Zhu, Q., Huff, J.: Hierarchical kohonenen net for anomaly detection in network security. IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 35(2), 302–312 (2005)
9. Jain, A., Dubes, R.: Algorithms for clustering data. Prentice-Hall, Inc., Englewood Cliffs (1988)
10. Lei, J., Ghorbani, A.: Network intrusion detection using an improved competitive learning neural network. In: 2nd Annual Conference on Communication Networks and Services Research, pp. 190–197 (2004)
11. Depren, O., Topallar, M., Anarim, E., Ciliz, M.: An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. Expert Systems with Applications 29(4), 713–722 (2005)
12. Bay, S., Kibler, D., Pazzani, M., Smyth, P.: The uci kdd archive of large data sets for data mining research and experimentation. SIGKDD Explor. Newsl. 2(2), 81–85 (2000)
13. DeLooze, L., DeLooze, L. A.F.: Attack characterization and intrusion detection using an ensemble of self-organizing maps. In: 7th Annual IEEE Information Assurance Workshop, pp. 108–115 (2006)
14. Mitrokotsa, A., Douligeris, C.: Detecting denial of service attacks using emergent self-organizing maps. In: 5th IEEE International Symposium on Signal Processing and Information Technology, pp. 375–380 (2005)

Transfer Graph Approach for Multimodal Transport Problems

Hedi Ayed, Djamel Khadraoui, Zineb Habbas, Pascal Bouvry, and Jean François Merche

-LITA, Universit Paul Verlaine -Metz
Ile du Saulcy 57045 METZ CEDEX 1
-CRP Henri Tudor, 29, Avenue John F.Kennedy
L-1855 Luxembourg-Kirchberg
hedi.ayed@tudor.lu
http://www.lita.univ-metz.fr/,
http://www.tudor.lu

Abstract. Route guidance solutions used to be applied to single transportation mode. The new trend today is to find route guidance approaches able to propose routes which may involve multi transportation modes. Such route guidance solutions are said to be multi modal. This document presents our contribution to multimodal route guidance problem. Following our strategy, we introduce a new graph structure to abstract multimodal networks. The graph structure is called transfer graph. A transfer graph is described by a set of (sub) graphs called components. They are connected via transfer points. By transfer point we mean any node common to two distinct components of a transfer graph. So a transfer graph is distinct from a partitioned graph. An example of transfer graph is a multimodal network in which all participating unimodal networks are not merged, but are kept separated instead. Since a multimodal network is reducible to a transfer graph, transfer graph based approach can be used for multimodal route guidance. Finally, to give meaning to our work, we try to insert our approach with the shortest path service in Carlink project. This step is seen as the implimentation of our algorithm, so we can get an idea on its performance.

Keywords: Multi modal transportation, multi objective optimization, time dependent network, graph theory, shortest path algorithm, route guidance.

1 Introduction

Multimodal transport, that is using two or more transport modes for a trip between which a transfer is necessary, seems an interesting approach to solving todays transportation problems with respect to the deteriorating accessibility of city centres, recurrent congestion, and environmental impact. One of this problems is to compute the shortest paht in a multimodal time dependent network.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 538–547, 2008.

Shortest path are one of the best studied network optimization problems (see e.g Bertsekas[3], Ahuja[1], and Schrijver[14]).

When investigating existing approches and algorithms on the topic, we observe that none of them is applicable to multi modal route guidance problems if subjects to the following constraints i) the underlying multimodal network is assumed to be flat and to contain no flat or hierarchical partitioning opportunity like regional hierarchy; ii) involved unimodal network may be kept separated and accessed separately; iii) if there are multiple network information sources within a single mode, they may be kept and accessed separately. They are simply disarmed. In fact relaxing these constraints is a precondition for them to work.

In the presentr work, a new approch for multi objective route guidance in time dependent multimodal network is proposed. Unlike previous proposals, it is able to compute multimodal route even if all involved networks are kept seperated and must be accessed separately. Instead of building a solution limited to the common view of multimodal route guidance, our strategy consist to find a fundamental network representation which covers multimodals networks and solve the shortest path problem on this abstract representation. This strategy has the advantage of providing us with a general tool fot solving the shortest path problem on any network reducible to the predefined abstract representation.

The present work has been done in the context of Carlink. Carlink is European project aiming to develop an intelligent wireless traffic service platform between cars supported by wireless tranceivers beside the road(s). The CARLINK system has to response to the necessity of providing an output or a service after occurring an event or receiving a request, interconnecting different modules and supporting different communications. The base on our system is Service Oriented Architecture (SOA). SOA is an evolution of distributed computing and modular programming. SOA builds applications out of software services. Services are relatively large, intrinsically unassociated units of functionality, which have no calls to each other embedded in them. They typically implement functionalities that most humans would recognize as a service, such as the information visualization or the warnings generation.

Our objective is to create a transport service to calculate the shortest path between two nodes of a multimodal multiobjectif network . This service can be used directly by a user or also by other services whatsoever Route "Service Planner," "Real Time Recommender road" or any other service. Such an application required the presence of a database describing the transport network (stations, roads, etc.) as well as information service providers of public transport (stops, tables times, etc.) This information is stored on servers Carlink and have the scope of the shortest path service.

2 Problem Description

Carlink route guidance service involves two basic functions: route planning and travel monitoring. Many constraints have been attached to Carlink route guidance service. First of all the service must be multi objective, i.e. user may specify

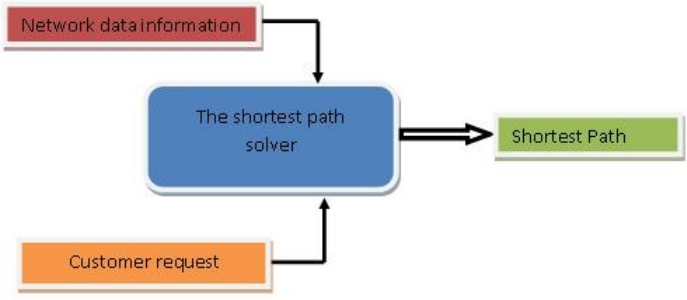


Fig. 1. Shortest Path Service interaction

a set of preferences and these preferences must be used to find optimal advisory. The service must also be multimodal, meaning that instead of computing paths base on a single transportation mode, paths are computed base a subset of at least two existing transportation modes (e.g. bus and rail). As a consequence and depending on user preferences, a path from a given origin to a given destination may pass trough nodes and links belonging to different transportation modes. A third constraint central to Carlink vision of multimodal route guidance, is the necessity to keep network information providers architecture as it is, i.e. distributed.

Though the finality is to build a multimodal route guidance system, the primary goal of the present work is to determine and implement algorithm(s) and approach(es) for multimodal route guidance. All the constraints above are likely to significantly determine design choices regarding them.

3 Classical Problems

Throughout this document, graph and related concepts will be frequently manipulated. The current section aims to identify graph concepts and provide some useful definitions. Classical shortest path problems in graph, as well as corresponding algorithms are also discussed. In the same vein, few words about multiobjective and time dependent shortest path problems are tod.

3.1 Classical Shortest Path Problem

With the above setting, the general formulation of the shortest path problem (SPP) can be stated as follow: given a graph G, a set of origin nodes O $\subseteq$ N and a set of destination nodes $D \subseteq N$, determine, for each node pair $s \in O$ and $t \in D$, a path p from s to t such that f(p) is minimal.

Depending on the size of O and D, there are four fundamental types of shortest path query: the *many-to-many* query ($|O| > 1$ and $|D| > 1$); the *many-to-one* query ($|O| > 1$ and $|D| = 1$); the *one-to-many* query ($|O| = 1$ and $|D| > 1$) and the *one-to-one* query ($|O| = 1 and |D| = 1$). Any shortest path problem falls into one of these categories. Each category raises a specific algorithmic challenge.

Except the last one which only require to add a "target reached" test as stop condition to the one-to-many shortest path algorithm.

Let (N, A) denote a given network, in which N is a set of n elements called nodes,$N = \{v_1, ..., v_n\}$ and A is a set of m elements called arcs $A = \{a_1, ..., a_m\} \subseteq N \times N$. Each arc $a_k \in A$ can be identified by a pair *(i,j)*, where *i, j* $\in N$. Each arc $a_k = (i, j)$ has associated a value, c_{a_k} or $c_{i,j}$, indicating the cost (or distance, time, etc..) to cross the arc.

3.2 Multiobjectif Shortest Path Problem

In many works on SPP, the path cost is a single scalar function. However there are some areas, like transportation, where one often needs to optimize path according to more than one scalar functions e.g. travel time and cost. In this case the underlying problem becomes a multi objective optimization problem. Some research efforts have been done to address bi-objective and exceptionally tri-objective shortest path. But concerning the demand of multi-objective optimization algorithms designed to accept a variable, and unbounded, number of objectives, the number of research effort is not that much. A SPP is said to be multi objective if the path cost function f maps to a real vector of dimension k ≥ 2, instead of a scalar. i.e. if $\mathcal{P}$ is the set of paths in a graph G, we have $f : P \rightarrow R^k, k \geq 2$, instead of $f : P \rightarrow R$. Each component $f_i \in f, (1 \leq i \leq k)$, follows the definition of single objective shortest path cost function.

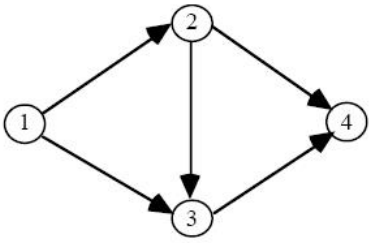
3.3 Time Dependent Shortest Path Problem

Let t denotes the time. In a time dependent network, each arc *(i,j)* is now associated with a time dependent arc cost function $c_{ij}(t)$ and a time dependent traversal delay function $d_{ij}(t)$. The arrival time at a node j after leaving a parent node i at time t is given by *arrival(t)* = t + $d_{ij}(t)$. When computing the SPT from a given root s, the tree $T_s(t)$, the potential $C_i(t)$ and the predecessor $\Pi_i(t)$ of a node i are all functions of time, but their expression may depend on the network time model. More details can be found in [10].

In the discrete time dependent network model, the time is not assumed to be continuous. Instead, an ordered set $T = \{t_1, t_2, ..., t_q\}$ of q discrete possible times is fixed, and for each arc *(i,j)* in the underlying graph, it is imposed that all departure time t, as well as all arrival time $t+d_{ij}(t)$ belongs to T. Each arc is also assigned a traversal delay for each time in T. The traversal delay of an arc *(i,j)* for different time in T is often described as follow: $d_{ij} = [d_1, d_2, ..., d_q]$. Figure 2 illustrates a discrete time dependent network model with $T = \{t_1, t_2, ..., t_{10}\}$. For simplicity, arc costs are assumed to always equal zero.

4 Transfer Graph Approach

According to the way they represent multimodal network and/or multimodal problem, we have identified three main approaches when studying existing works



$d_{12}(t) = [1, 1, 3, 3, 3, 3, 3, 3, 3, 3]$

$d_{13}(t) = [2, 3, 3, 3, 3, 3, 3, 3, 3, 3]$

$d_{23}(t) = [1, 1, 1, 1, 2, 2, 2, 2, 2, 2]$

$d_{24}(t) = [1, 1, 3, 3, 2, 2, 2, 2, 2, 2]$

$d_{34}(t) = [1, 1, 1, 1, 2, 2, 2, 2, 2, 2]$

Fig. 2. A time dependent network with arcs traversal delay

on multimodal route guidance: the multigraph based approach (see [5], [6], [8], [9], [11], [13]), the constraint satisfaction problem (CSP) based approach (see [2] and [4]), and the grid based approach (see [7]).

From our point of view, current approaches are simply not appropriated to satisfy at least the three first multimodal route guidance characteristics listed above, and as far as we know, no ready to use theoretical tool is currently available for that. In this section we propose an approach which sets the basis to solve the problem as specified. The discussion starts with a description of our approach, followed by a formal description of transfer graph. Next we analyze the shortest path problem in transfer graph. We then propose algorithms for basic version of the problem. These basic algorithms will be revised later to solve multi objective time dependent shortest path problem in transfer graph.

4.1 Approach Description

The most constraining requirement we have to manage is to compute multimodal routes while keeping all existing unimodal transportation networks separated. To abstract the particular multimodal network configuration imposed by this requirement, we will introduce an unusual graph structure which we call transfer graph. In few words, a transfer graph is described by a set of graphs or components and a set of transfers connecting them. Figure 3 illustrates how a transfer graph looks like. The concept is well described in sections ahead where a basic formalization is given. For the moment it is enough to observe that each graph G1, G2, G3 is a transfer graph component. There are three transfers. A transfer connects two components via transfer points, a transfer point being a node common to the two components involved in the transfer. Observe that source (the dark node) and destination (the doted node) may belong to more than one component.

Let *G=(N, A)* denotes a graph. G may be decomposed into a set $GS = \{G_1, G_2, \cdots, G_q\}$ of sub graphs. Each $G_g = (N_g, A_g)$ is called a component and is such that $N_g \subset N$ and $A_g \subset A$. GS itself is such that $N = \bigcup_{N_g \in GS} N_g$ and $A = \bigcup_{A_g \in GS} A_g$. Unlike partitioned graph, given two distinct components G_g, $G_{g'}$, having $N_g \cap N_{g'} = \varnothing$ is not mandatory. However $A_g \cap A_{g'} = \varnothing$ must always hold.

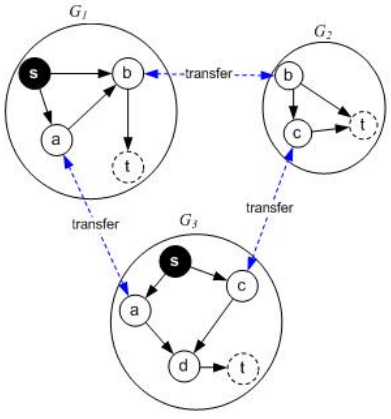


Fig. 3. Transfer graph illustration

Again, let two distinct components G_g, $G_{g'} \in GS$. If $N_g \cap N_{g'} \neq \varnothing$, then any $i \in N_g \cap N_{g'}$ is a transfer point. In other words a transfer point is a node which belongs to more than one component. To express that G_g, $G_{g'}$ are connected via node $i \in N_g \cap N_{g'}$, we use structure *(g, g′, i)* or *(g′, g, i)* which denote transfer between components. More literally a transfer provides the following information: "It may be possible to move from component G_g to component $G_{g'}$ via node i". If transfers *(g, g′, i)* and *(g′, g, i)* are not equivalents, then they are directed. From here on, we assume that transfers are not directed.

For a decomposition GS of graph G, we write $TS = \{\tau : \tau = (g, g', i) \wedge G_g, G_{g'} \in GS \wedge i \in N_g \cap N_{g'} \neq \varnothing\}$ to denote the transfer set , i.e. the set of all transfers derived from GS. Normally, TS is computed/derived from GS. But it may be explicitly specified, especially in case some additional properties are attached to each transfer. If TS $= \varnothing$, it means that there is no connection between components. Two components $G_g, G_{g'} \in GS$ being connected iff $(g, g', i) \in TS$ or $(g', g, i) \in TS$.

So what is a transfer graph? We denote a transfer graph by the structure TG=(N, A, GS, TS) where N is the set of all nodes; $A = \{(i, j) : i, j \in N\}$ is the set of arcs; GS is a decomposition of the graph described by (N, A) as specified above; and TS is the transfer set. From here on, to denote that an element or an element feature x is viewed from a given component G_g, we will write x^g. So a vertex from component G_g will be denoted by i^g, arc will be denoted by $a^g = (i, j)^g$ and a feature f of arc $(i, j)^g$ will be denoted by f_{ij}^g .

Given a transfer graph TG, one may be interested by many features:

- Given a transfer graph node $i \in N$, the node containers of i is the set of components containing a node i. It is denoted by $NG_i = \{G_g : (g, g', i) \in TS \vee (g', g, i) \in TS\}$.
- the set of transfer in which G_g is involved, we call it component transfer set and denote it by $TS_g = \{\tau : \tau = (g, g', i) \vee (g', g, i) \vee \tau \in TS\}$

– the set $TP = \{i : (g, g', i) \in TS\}$ of all transfer points in a given transfer graph or by the set $TP_g = \{i : (g, g', i) \in TS \vee (g, g', i) \in TS_g\}$ of all transfers points within a transfer graph component G_g.

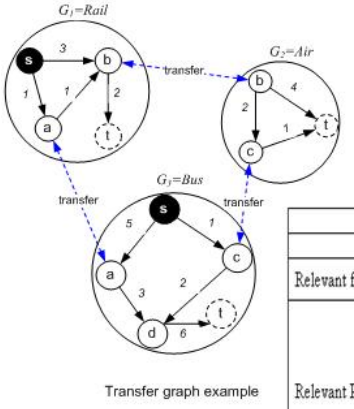
4.2 Shortest Path Algorithm in Transfer Graph

Consider a transfer graph T*G=(N, A, GS, TS)*. Let s, $t \in N$ be an origin-destination pair and $G_g \in$ GS be a component of TG. At a high level, paths in a transfer graph can be divided into two groups: *intra components* paths and *inter component paths*. An intra component path within G_g is any path which connects two nodes $i, j \in N_g$ while traversing only arcs belonging to G_g. On the other hand, an inter component path within a transfer graph TG is any path which connects two nodes $i, j \in N$ while traversing arcs from at least two distinct components. Intra component paths can be subdivided into two subcategories: *full paths* and *partial paths*. A full path is a path which connects source to target, while a partial path is any non empty path which is not a full path. Partial paths fail into three sub categories: *head paths*, *tail paths* and *intermediate paths*:

– **Full paths :** An intra component full path is a path which connects a source node s to a target node t. It is of the form $p = \langle s, a_0^g, v_1, a_1^g, v_2 \cdots, t\rangle$.
– **Relevent Head paths :** An intra component head path is a path which starts from the source s and ends with any node different from target t. Head paths are of the form $p = \langle s, a_0^g, v_1, a_1^g, v_2 \cdots, x\rangle$ with $x \neq t$.
– **Relevant intermediate paths:** An intra component intermediate path starts from and ends with any node different from s and t. Intermediate paths are of the form $p = \langle x, a_0^g, v_1, a_1^g, v_2 \cdots, y\rangle$ with $x, y \notin \{s, t\}$.
– **Relevant tail paths:** An intra component tail path is a path which starts from any node different from the source s and ends with target t. Tail paths are of the form $p = \langle x, a_0^g, v_1, a_1^g, v_2 \cdots, t\rangle$ with $x \neq s$.

Given a transfer graph TG=(N, A, GS, TS), and an origin-destination pair. s, $t \in N$, assume that for all components $G_g \in$ GS we have computed the following relevant path sets: $P_{s.t}^{*g}$(the set of best intra component full path within G_g), $P_{s.-}^{*g}$(the set of all best intra component head paths from s within G_g), $P_{+.t}^{*g}$ (the set of all best intra component tail paths to t within G_g) and $P_{+.-}^{*g}$ (the set of all best intra component intermediate paths within G_g).

Having these relevant path sets in hand, it is possible to derive a special graph from which all possible best inter component paths from s to t can be found. We call this graph *the relevant graph* and use *RG* to denote it. The node set of RG is a sub set of all intra component nodes. The arc set of RG is the set of all computed relevant intra component partial paths viewed as edges. In fact RG is a multigraph, but in general, its scale should faraway be smaller than that of the equivalent multigraph of the underlying transfer graph. This is because except origin and destination nodes, any other node appearing in RG iff it is a transfer point. So the number of nodes in RG directly depends on the number of transfer points in the whole transfer graph. In order to formally describe *RG*,



Relevant intra component paths in *G1*			
Path types		Paths	Weight
Relevant full path		$\langle (s,a)^1,(a,b)^1,(b,t)^1 \rangle$	4
Relevant Partial paths	*Head paths*	$\langle (s,a)^1,(a,b)^1 \rangle$	2
		$\langle (s,a)^1 \rangle$	1
	Intermediate paths	$\langle (a,b)^1 \rangle$	1
	Best tail paths	$\langle (a,b)^1,(b,t)^1 \rangle$	3
		$\langle (b,t)^1 \rangle$	2

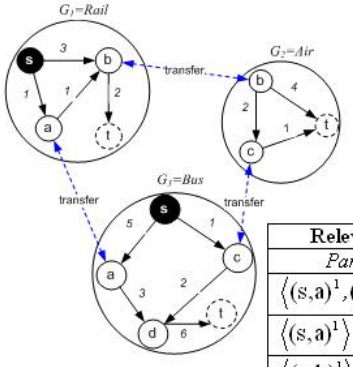
Fig. 4. Intra component paths illustration

let $RV^g = TP_g$ denotes the set of relevant nodes in component G_g. RV^g is simply the set of transfer points in G_g . Let RE^g be the set of edges representing relevant paths computed in component G_g.

From all RV^g and RE^g, we can respectively build $RV = (\bigcup_{G_g \in GS} RV^g) \bigcup \{s, t\}$ the set of relevant nodes from all component G_g, and $RE = \bigcup_{G_g \in GS} RE^g$ the set of relevant paths representative edges from all component $G_g \in GS$ The relevant graph is formally described by tuple *RG=(RV, RE)*. If the transfer graph is connected, then RG is also a connected graph. Otherwise, RG may not be connected. Figure 5 shows the relevant edge set RE^1 computed from component G_1 of the associated transfer graph.

Now assume we have built the relevant graph RG of transfer graph *TG=(N, A, GS, TS)* given an origin-destination pair $s, t \in N$. With this setting, computing the shortest path from s to t in RG is simply the classical origin-destination shortest path problem. Let $P_{s.t}^{RG}$ denotes the set of all possible inter component full paths from s to t in RG, and let $P_{s.t}^{*RG}$ be the set of all best inter component full paths from s to t in RG. Running a single source shortest path algorithm on RG will return a path $p \in P_{s.t}^{*RG}$ passes through more than one component. When considering a component G_g, it is easy to observe that the computation of each kind of relevant best paths in G_g from source s to target t corresponds to a variant of the shortest path problem. More precisely it is easy to observe that:

- $P_{s.t}^{*g}$ (the set of best full paths in G_g) corresponds to the set of solutions of a standard ***one-to-one*** shortest path algorithm within G_g;
- $P_{s._}^{*g}$ (the set of best head paths in G_g) corresponds to the set of solutions of a standard ***one-to-many*** shortest path within G_g, with destination nodes set being the set of outgoing transfer points within G_g except t;



Relevant path-edges from component *G1*		
Partial path	*Path-edge*	*Weight*
$\langle (s,a)^1,(a,b)^1 \rangle$	$(s,b)^1$	2
$\langle (s,a)^1 \rangle$	$(s,a)^1$	1
$\langle (a,b)^1 \rangle$	$(a,b)^1$	1
$\langle (a,b)^1,(b,t)^1 \rangle$	$(a,t)^1$	3
$\langle (b,t)^1 \rangle$	$(b,t)^1$	2

Fig. 5. Relevant path-edges from a transfer graph component

- $P^{*g}_{+.t}$ (the set of best tail paths in G_g) corresponds to the set of solutions of a standard ***many-to-one*** shortest path within G_g, with origin node set being the set of incoming transfer points within G_g except s;
- $P^{*g}_{+.-}$ (the set of best intermediate paths in G_g) corresponds to the set of solutions of a standard ***many-to-many*** shortest path within G_g, with origin node set being the set of incoming transfer points within G_g except s and t, and the destination node set being the set of outgoing transfer points within G_g except s and t.

5 Conclusion

The work presented in this document has been done within multimodal route guidance work package of the Carlink project. The goal was to investigate algorithms and approaches in multimodal route guidance. Among other characteristics, the targeted solution was expected to support multi objective route guidance in time dependent multimodal network. Our mission was to dig into multimodal route guidance problem, to propose a solid foundation on top of which Carlink's multimodal route guidance system will be built and to materialize our proposal in an extensible route guidance library.

References

1. Ahuja, R., Mananti, T., Orlin, J.: Network Flows: Theory, Algorithms, and Application. Prentice Hall, Englewood Cliffs (1993)
2. Goldberg, A.V., Harrelson, C.: Computing The Shortest Path: A* Search Meets Graph Theory Technical Report MSR-TR-2004-24 (March 19, 2003)

3. Bertsekas, D.: Linear Network optimization: Algorithms and Codes. MIT Press, Cambridge (1991)
4. Chiu, D.K.W.W., Lee, O.K.F., Ho, H.-f.-f.L., Au, E.W.K., Wong, M.C.W.: A Multi Modal Agent Based Mobile Route Advisory System for Public Transport Network. In: Proceedings of the 38th Hawaii International Conference on System Sciences - 2005 (2005)
5. Meng, F.H., Yizhi, L., Chuin, L.H.W.L.H.: A Multi-Criteria, Multi-Modal Passenger Route Advisory System, http://citeseer.ist.psu.edu/281757.html
6. Zidi, K.: Systéme Interactif d'Aide au Déplacement Multimodal (SIADM)"; Thése préparée dans le laboratoire LAGIS á l'Ecole Centrale de Lille et l'Université des Sciences et Technologies de Lille (Décembre 13, 2006), http://tel.archives-ouvertes.fr/tel-00142159/en/
7. Fragouli, M., Delis, A.: EasyTransport: An Effective Navigation and Transportation Guide for Wide Geographic Areas. In: 14th IEEE International Conference on Tools with Artificial Intelligence, 2002 (ICTAI 2002) Proceedings, pp. 107–113 (2002)
8. Bielli, M., Boulmakoul, A., Mouncif, H.: Object modeling and path computation for multimodal travel systems. European Journal of Operational Research
9. McCormack, J.E., Roberts, S.: Exploiting Object Oriented Methods for Multimodal Trip Planning Systems Report 94.7 (April 1994)
10. Pallotino, S., Scutella, M.G.: Shortest Path Algorithms in Tr ansportation models: classical and innovative aspects, Technical Report: TR-97-06 (1997)
11. Li, Q., Kurt, C.E.: GIS-Based Itinerary Planning System for Multimodal and Fixed-Route Transit Network. In: Mid-Continent Transportation Symposium Proceedings
12. Daruwala, R.-S.: On computing the Pareto optimal solution set in large scale dynamic network. A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Departement of Computer Science New York University (September 2002)
13. Kitamura, R., Chen, N., Chen, J.: Daily Activity and Multimodal Travel Planner Final Report; Year 1999 Paper UCB-ITS-PRR-99-1
14. Schrijver, A.: Combinatorial Optimization. Polyhdra and Efficiency. Springer, Heidelberg (2003)

Wireless Traffic Service Communication Platform for Cars

Timo Sukuvaara[1], Pertti Nurmi[1], Daria Stepanova[1], Sami Suopajärvi[1], Marjo Hippi[1], Pekka Eloranta[2], Esa Suutari[3], and Kimmo Ylisiurunen[4]

[1] Finnish Meteorological Institute, Erik Palménin aukio 1, FIN-00560 Helsinki, Finland
timo.sukuvaara@fmi.fi
[2] Mobisoft Oy, Hatanpään valtatie 26, FIN-33610 Tampere, Finland
[3] Sunit Oy, Kehräämöntie 4, FIN-87400 Kajaani, Finland
[4] Infotripla Oy, Kehräsaari A, FIN-33200 Tampere, Finland

Abstract. Rapidly changing weather conditions, especially in winter, have caused numerous disastrous traffic accidents in Northern Europe and in the Alpine region during recent years. Information about hazardous weather and road conditions is often potentially available but difficult or sometimes even impossible to deliver to drivers. This paper presents the international CARLINK (Wireless Platform for Linking Cars) project [1] of the Celtic Cluster Programme Call 3 whose aim is to develop an intelligent wireless traffic service platform between cars supported with wireless transceivers along the roads. The platform consists of a specific set of services, however not only these but variety of other services can be integrated to this kind of a system. Two of the major services are real-time local road weather service and incident warning service. The real-time local road weather service is a solution where up-to-date local weather related information is being collected from cruising vehicles and then further delivered to other vehicles in the area. Incident warning service operates in the same manner, but concentrates to the parameters related to traffic incidents or accidents, and (depending on seriousness of the incident) delivers a warning of such events to the vehicles in the traffic network without delay. The ultimate goal is to develop an intelligent communication platform for vehicles so that they can deliver their own observations of traffic and weather conditions to the platform core.

Vehicular networking is nowadays a widely studied research field, and a large number of suggestions for vehicle-to-vehicle and vehicle-to-infrastructure communications have been presented. The focus is typically on bilateral communication between two vehicles or on broadcasting information from one vehicle or infrastructure to vehicles in the surrounding area. The CARLINK project is developing an intelligent hybrid wireless traffic service platform between cars supported with wireless base stations beside the road(s). Communication between the cars will be arranged in an ad-hoc manner, supported with a wireless base station connection to the backbone network, whenever possible. The ultimate goal is to enhance traffic safety and smoothness, but also to generate completely new communication entity, allowing new types of applications, services and business opportunities. Not only the encountering cars and

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 548–557, 2008.

the infrastructure can broadcast data, but all the data can be delivered instantly over the communications network to all CARLINK-compliant vehicles. High impact and extreme weather generated challenges are increasing throughout the world, not least because of the climate change. CARLINK can truly contribute to meeting these challenges. The preliminary network simulations, communication tests and weather service prototypes have already shown that a new kind of wireless communication environment can be created and it is indeed capable of enhancing traffic safety.

1 Introduction

Car-to-car communication platform is currently a popular research topic, having various different approaches. A common factor for this kind of networking topology is Vehicular MANET (VANET), where MANET stands for Mobile Ad-Hoc Networks. One of the major activities in this area is the Car-to-Car Communication Consortium (C2C-CC) driven by the major European car manufacturers and aiming at generating decentralized floating car data (FCD) communication capabilities between cars [2]. The objective in C2C-CC is to provide mainly broadcast-type of services, such as broadcasting accident warnings from car-to-car and roadside information such as intersection guidance from traffic infrastructure to car. C2C-CC concentrates mainly on services and applications. In telecommunications the aim is to support the standardization activities driven by IEEE (WAVE - IEEE 802.11p, IEEE 802.11 a/b/g) [2,3]. Similar approach where cars distribute accident warning data from car-to-car and even forwarding warnings car by car in ad-hoc networking manner was presented in 2006 in the IEEE Communications Magazine [4]. The other example is the LIWAS traffic warning system [5], designed to provide early warnings to vehicles about adverse road conditions like a slippery road surface. The LIWAS system is currently under development and will consist of two major parts: (a) sensors for determining the state of the road, and (b) communication infrastructure supporting inter-vehicle communication. The most of the European activities in this area are more or less related to the C2C-CC work, as well as to the e-Safety initiative [6] of the European Union and the EU's COMeSafety project. The most popular wireless high-speed communication approaches are nowadays Wireless Local Area Networks (WLAN) also known as Wi-Fi (Wireless Fidelity), and WiMAX. WLAN is based on the IEEE 802.11 standard family. The most common versions nowadays are the 802.11b and 802.11g standards operating in the 2.4 GHz bandwidth and capable of up to 54 Mbps (.11g) or 11 Mbps (.11b) data speeds, respectively. The WLAN standards support a moderate level of mobility, e.g. users moving at walking speed.

The IEEE 802.16 family of standards specifies the air interface of both the fixed and the mobile broadband wireless access (BWA) systems for supporting multimedia services. The WiMAX system is based on these technologies and is sponsored by an industry consortium called WiMAX Forum. IEEE 802.16-2004 for fixed and IEEE 802.16e for mobile access, respectively, are the IEEE standards which define the current structures of the WiMAX system. WiMAX has licensed worldwide

spectrum allocations in the 2.3 GHz, 2.5 GHz, 3.3 GHz and 3.5 GHz frequency bands and is capable of up to 31.68 Mbps data rates with a single antenna system and up to 63.36 Mbps with a multiple antenna system. The WiMAX system is capable of supporting fast moving users in a mesh network structure. Systems with users moving at speeds up to 60 km/h have been reported [7].

The IEEE standardization activity for the car-to-car communication environment is named as WAVE (IEEE 802.11p) [3]. The underlying technology in this standardization work is called Dedicated Short-Range Communication (DSRC), which is essentially the IEEE 802.11a standard adjusted for low overhead operations. The primary purpose of the DSRC is to enhance public safety applications, to save lives and to improve traffic flow by vehicle-to-vehicle and infrastructure-to-vehicle communications. In the U.S. the 75 MHz channel is allocated for the DSRC in the 5.9 GHz spectrum [8].

Another related area of research is the service-oriented approach with a goal to improve traffic safety and comfort. Weather conditions in winter, especially when rapidly changing, are a reason behind numerous disastrous traffic accidents in Northern Europe and in the Alpine region during the recent years. Information about hazardous weather conditions is often potentially available but difficult or sometimes even impossible to deliver to the drivers in the area. A tragic example of such an incident is the chain collision of cars nearby Helsinki in March 2005 in the morning, where 3 persons lost their lives and tens of people were injured. Although the hazardous driving conditions were forecasted by the Finnish Meteorological Institute already a day before the accident, several accidents took place in a rather small geographical area. Temperature was below -10 C degrees and the surface changed suddenly to very slippery because of light snow. There were no methods to deliver road condition and accident information to all vehicles and thus to prevent the accidents. Later on, several type of solutions have been designed for the traffic safety improvement. One example is the Finnish national “VARO” project where safety is being improved by delivering warnings and route guidance to the end-user devices located in cars. A similar approach is the provision of traffic congestion information to car navigation tools. Such systems have been developed into various navigation equipments and are already available in several countries. One approach to provide traffic services is to equip a mobile handheld terminal with a tranceiver being able to receive broadband data. Traffic and accident data can be obtained directly over the Internet with this equipment. The Celtic Wing TV project is researching this scenario, relying on the DVB-H broadcasting standard. Similar studies are ongoing by mobile communication device manufacturers worldwide.

In the CARLINK project the aim is to build more comprehensive solution for car networking, car to car communication purposes and traffic safety improvement. An intelligent hybrid wireless traffic service platform between cars has been developed, supported with wireless transceivers acting as access points along the roads. The ultimate goal of this concept is to enhance traffic safety, as well as to allow communication between cars and between cars and common communication infrastructure. The simulations and analysis evaluate the communication efficiency of the platform.

2 Platform and Services

The CARLINK wireless traffic service platform is designed to provide an infrastructure to a wide community of commercial and governmental traffic and safety services. It is a wireless ad-hoc type communication entity with connectivity to the backbone network via base stations. The platform itself is the key element of CARLINK, but the services created to the platform have also a crucial role; On one hand, they generate different ways to use and to exploit the platform, proving it efficiency. But on the other hand the services are the showcase of the platform towards the consumers; in order to make consumers interested in purchasing the platform (and furthermore vehicle industry to integrate the platform equipment to the vehicles) there is a need to have some key services interesting and necessary enough for the consumers. CARLINK is not planning to build up an extensive package of services, but just a couple of key services to prove the applicability, usefulness and necessity of the platform, and so-called "killer-application" to raise the public interest.

CARLINK has defined the example set of services for the platform listed in the Table 1. The local road weather service (RWS) collects observed weather data from vehicles and Traffic Service Base Stations (TSBS) which are installed by the road weather stations and use these observations together with the weather information from other sources to generate comprehensive precise local road weather analysis and forecasts to be forwarded back to the cars. The cars measure air temperature and TSBSs measure e.g. road surface and air temperature, state of the road and precipitation. The incident/emergency warning service uses vehicle data to generate warnings related to exceptional traffic conditions or accidents. The traffic service will generate traffic logistics data for the public authorities. Finally, the remaining services listed deliver commercial-like travel data to users on the move. In this paper we concentrate on the local road weather service and the incident/emergency warning service, since they together exploit most widely the capabilities of the CARLINK platform, enhance traffic safety, and based on the CARLINK partner expectations form the "killer-application" of the platform.

The Wireless Traffic Service Platform is divided into three parts: Traffic Service Central Unit (TSCU), the base station network with Traffic Service Base Stations (TSBS), and Mobile End Users (MEU) with ad-hoc connectivity and (non-continuous) backbone network connectivity.

The overall structure of the platform is presented in Figure 1. The MEUs form a wireless network. They do not have continuous connectivity but operate in ad-hoc manner with each other whenever possible, typically when two cars pass each other.

Always when a vehicle with a MEU passes a TSBS, it will get up-to-date traffic platform information stored into the TSBS. The TSBS receives regular updates to the traffic platform information from the TSCU, located in the fixed network beyond the TSBS. The TSCU operates in the fixed network relying exclusively on the existing communication solutions of the fixed networking. The TSBS acts as an interface between the fixed and wireless networks. However, the MEU also

Table 1. CARLINK services

Service name	*Brief Description*
Transport	Transport guidance and real-time timetables
Traffic	Traffic logistics for traffic control centre
Local Road Weather	Up-to-date local weather information for vehicles
Positioning	Vehicle positioning
Route planner	Planning route to expected destination
Parking places	Real-time parking place availability info
Point of Interest	Guidance to point of interest
Geo-coding	Geometric data
Incident warning	Instant warning of accidents and incidents ahead

transmits data to/from TSCU over the lower capacity (GPRS) connection when critical weather, warning or accident information emerge.

The operation in the local RWS and the warning service are uniformly constructed of the procedure presented in the Figure 2. The TSCU maintains up-to-date local road weather information and forwards it regularly to the TSBSs in the area of interest/vicinity area. Each TSBS has therefore up-to-date local road weather information, which is delivered to every MEU passing by the TSBSs. The MEU receives and applies the weather data and in exchange it forwards the collection of its own weather and traffic related measurements. This data is delivered back to the TSCU where it is used to update the local road weather data and to generate potential additional warnings. The MEUs are also exchanging data during the encounters: the MEUs deliver their own up-to-date data to the other MEUs, and the more recent data will be used by all. In the case of emergency, the two-layer networking procedure can be bypassed with the parallel direct reliable low-capacity mobile phone network based communication between the TSCU and the MEU. This channel may not be adequate for the full scale data but, due to its practically complete coverage, critical emergency data is delivered without delay.

The local RWS is derived from FMI's (Finnish Meteorological Institute) road weather model [9] presented in the Figure 3 is a one-dimensional energy balance model which calculates vertical heat transfer in the ground and at the ground-atmosphere interface, taking into account the special conditions prevailing at the road surface and inside the ground below. The model also accounts for the effect of traffic volume on the road. The output from a Numerical Weather Prediction (NWP) model is typically used as a forcing at the upper boundary. This information provides also the horizontal coupling between individual computational points of the model. The basic horizontal resolution of the FMI's present Road Weather Model is as sparse as 10 km which means that in principle the model cannot resolve the meteorological features beyond this spatial scale. The main body of calculations relate to the conditions within the ground, where the vertical temperature distribution is solved to a depth of down to c. six meters. The model atmosphere is considered as a forcing factor having an effect on the ground surface through a number of variables like ambient temperature, relative

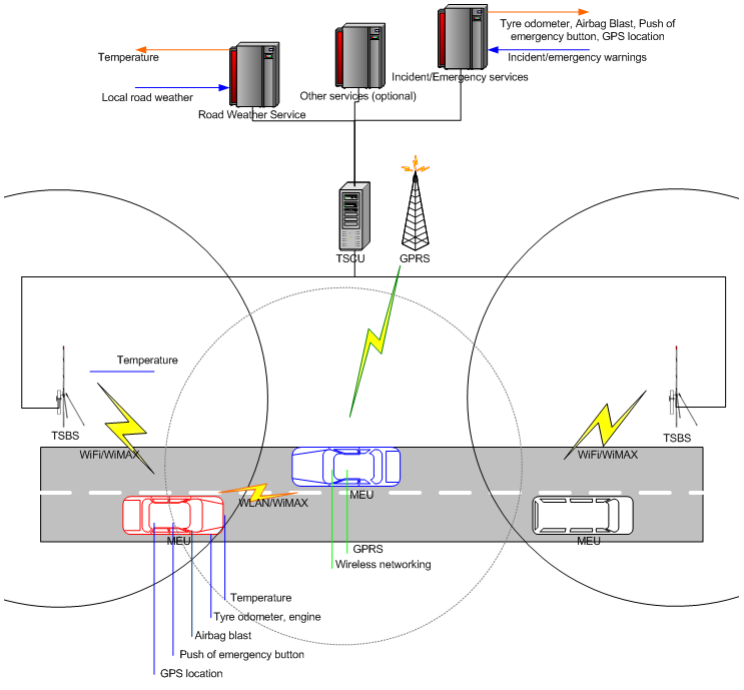


Fig. 1. CARLINK platform structure

humidity, wind speed, short- and long-wave radiation, and precipitation. The values of these variables can be inferred from observations or from a forecast, i.e. the model does not make a distinction as to the source of the input data. The heat balance at the ground surface is solved on the basis of these variables and taking into account such additional factors as sensible and latent heat fluxes as well as atmospheric stability. The effect of melting and freezing is also included in the energy balance.

An additional forcing at the surface is the traffic volume, which causes not only increased turbulence but also mechanical wear of e.g. snow, ice or frost that prevails on the surface. A spatially constant traffic effect is assumed in the model, and during the night time a smaller traffic factor is used. Further to calculating the ground and road surface temperatures, the model performs a road condition interpretation. Eight different forms of road surface description are used: dry, damp, wet, frost (deposit), dry snow, wet snow, partly icy, and icy. The model furthermore combines information of the road conditions, storage sizes and certain weather parameters to produce a three-valued traffic condition index describing the traffic conditions in more general terms. They are: normal, bad, and very bad, and this same classification is used for traffic condition warnings issued by FMI.

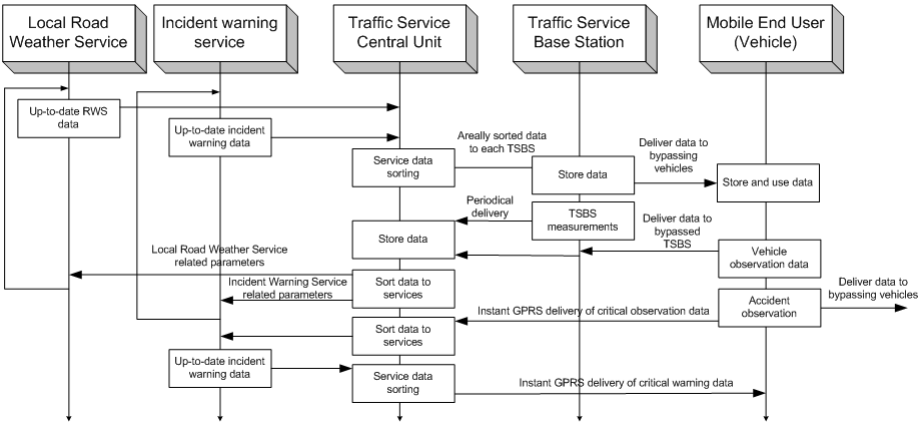


Fig. 2. Operational model of Local RWS and Incident warning service

3 Technical Requirements

The platform services presented in the previous chapter set several requirements for the CARLINK communication system. Networking challenges can be roughly divided into fixed networking between the TSCU and the TSBS, communication via the wireless base station between the TSBS and the MEU, wireless ad-hoc communication between the MEUs, and reliable low-capacity communication between the TSCU and the MEU, respectively.

The existing fixed networking methods provide service levels which clearly fulfill the CARLINK requirements for the communication between TSCU and the TSBS. For the communication between the TSBS and the MEU, ordinary communication via a wireless base station without a need to hand off the connection but with one extremely challenging element, the traffic speed (up to 100 km/h in our scenario), is required. The most popular solution for wireless communication is the WiFi system which is based on the IEEE 802.11 standard family. The latest version of the standard is the IEEE 802.11g standard, capable of 54 Mbps data speed and with a coverage at least up to 100 meters (maximum range allowing only 1 Mbps data speed). The use of this system at traffic speeds is a challenging task. The time a vehicle stays in the area of the base station is rather short for initiating the connection and carrying out data exchange. Also the IEEE 802.11g standard is not especially developed for the network of high-speed nodes, and the Doppler effect as well as fast-changing received power level may decrease the performance even more. The 802.11 standardization forum has noted that existing 802.11 standards (a,b,g) are not optimal solutions for fast nodes, and is tackling the issue of vehicular communication especially in the 802.11p standardization work, based on IEEE 1609 standard family. The WiMAX (Mobile WiMAX IEEE 802.16e) method is clearly more suitable for this kind of networking due to its better coverage. However, neither the mobile WiMAX based on IEEE 802.16e nor IEEE 802.11p components are not yet

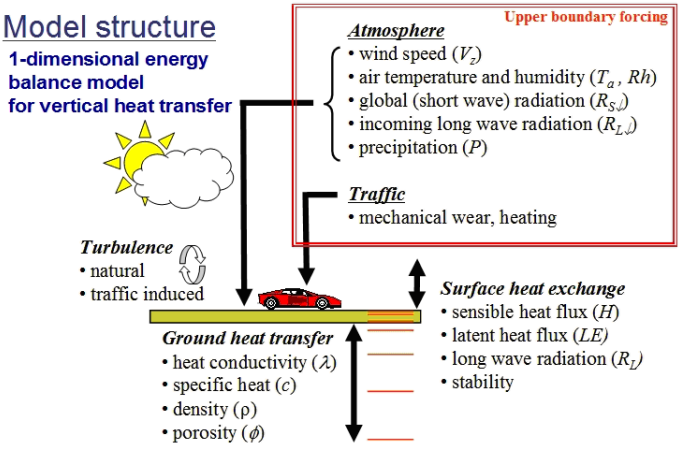


Fig. 3. Schematic of the road weather model

available in large scale. In the CARLINK research WiFi based on IEEE 802.11g stands for existing communication product, while mobile WiMAX represents interesting design alternative of emerging technologies. In this paper we have analyzed 802.11g in the simulations developed in the NS-2 tool and in the simple test system based on the WiFi (IEEE 802.11b) wireless communication to find out the possible constraints (like the Doppler effect caused by a fast moving vehicle) and to prove the concept operability.

The technical challenges in the ad-hoc communication between the vehicles are basically the same as in the base station-end-user communication except that the encounter speed is doubled (and the communication time halved) because in the extreme situation both counterparts are moving into the opposite directions (towards each other). Hence, it is extremely hard to enable full scale ad-hoc networking. In order to ensure platform operability the MEUs only to be able to exchange their packed up-to-date data in encounter, not to allow the true ad-hoc networking are required.

Finally, it is essential to ensure that the most crucial data will be exchanged between the extreme parts of the network (from the MEU to the TSCU and back) without any delay. For this purpose a standard GPRS data service developed for the GSM mobile phone network is used. This solution guarantees the reliability required in this particular scenario, even if the capacity may be too low for the full-scale platform service.

4 System Description

Based on the technical requirements presented above, CARLINK has created the platform structure illustrated in the Figure 1. On the top of the platform there is the TSCU with connections to the underlying service cores, the local

traffic weather service and the incident/emergency warning service. The TSCU takes care of the user management. As a central unit of the system, the TSCU is maintaining the interdependencies of all the platform elements. It also stores all the data gathered from the platform and forwards the appropriate data both to the road weather service (RWS) and to the incident/emergency warning service. In the incident/emergency warning service platform there is the TSCU with connections to the underlying service cores, the road weather service and the incident/emergency warning service.

The incident/emergency warning service parameters are an airbag blast, a push of the emergency button in the car, a tyre odometer and an engine status, all of them including the GPS-position of the observed issue. The RWS core includes a weather forecast model generating a local road weather outlook based on the FMI's operational measurements. This model is supplemented with car measurements (temperature and GPS-position of observations) to complement the weather information. The resulting local road weather information is delivered to the TSCU which is responsible for forwarding this data to the vehicles through the CARLINK platform. Similarly, the incident/emergency warning service collects vehicle data to build up warnings with these exact locations, returned to the TSCU. Depending on the significance of the warning the TSCU selects the appropriate path for the warning data distribution. The most critical warnings (e.g. accident location) are delivered through the GPRS connection as rapidly as possible, while the more informative-like warnings can be distributed through the base stations.

There is a network of TSBSs below the TSCU (Figure 1), mainly acting as a data transmitter from the TSCU to the MEUs and vice versa. The TSBS is also collecting weather data itself and delivering it to the TSCU.

The MEUs in vehicles are the users of the CARLINK platform. They are gathering raw platform data along the roads they are driving, delivering data up to the TSCU and the underlying service cores and, finally, consuming the weather and warning information (partially) derived from the vehicle based data. The parameters gathered from the vehicle are the temperature, combined tyre odometer and car gyroscope information, airbag blast notification, push of emergency button notification and the GPS position for each data source. The gyroscope, the tires and the GPS-system each have their own interfaces, while the push of the emergency button will be gathered from the drivers user interface. The remaining parameters (temperature, airbag blast notification) are coming from the vehicle CAN-bus gateway. The WLAN/WiMAX and the GSM/GPRS interfaces are used for the communication with the TSBSs and the TSCU.

5 Conclusions

This paper has presented the CARLINK concept of an intelligent hybrid wireless traffic service platform between cars, supported with wireless transceivers acting as access points along the roads. The ultimate goal is to create an intelligent communication platform for vehicles where they can deliver their own observations

of traffic and weather conditions to the platform core. This information is delivered back to the vehicles as analyzed (and forecasted) information about road weather conditions and as immediate incident warnings. Compared to car-to-car or infrastructure-to-car solutions presented by the car industry the CARLINK solution showcases a true bidirectional communication entity. Within CARLINK not only the encountering cars, or the encountering car and the infrastructure, can broadcast data, but all data can be delivered instantly through the network to all CARLINK-compliant vehicles. The solution has been presented on concept level, and the further work will cover comprehensive simulations and a system test approval of our concepts.

Acknowledgments. This work has been supported in part by the Technology Advancement Agency of Finland (TEKES) and the European Union Eureka cluster program Celtic. The authors wish to thank all our partners of the CARLINK project.

References

1. Sukuvaara, T., Stepanova, D., Nurmi, P., Eloranta, P., Suutari, E., Ylisiurunen, K.: Wireless Traffic Service Communication Platform for Cars. In: 2nd IEEE Workshop on Automotive Networking and Applications (AutoNet 2007) Co-located with IEEE GLOBECOM 2007, Washington, DC, USA, November 30 (2007)
2. Kosch, T.: Technical Concept and Prerequisites of Car2Car Communication. In: 5th European Congress and Exhibition on ITS, Hannover, Germany (June 2005)
3. IEEE 802.11p Wireless Access for Vehicular Environments, Draft Standard
4. Biswas, S., Tatchikou, R., Dion, F.: Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety. IEEE Communications Magazine 44(1), 74–82 (2006)
5. Broensted, J., Hansen, K.M., Kristensen, L.M.: An infrastructure for a traffic warning system. In: International Conference on Pervasive Services, ICPS 2005 Proceedings, pp. 136–145 (2005)
6. Strategic research agenda – ICT for mobility, eSafety Forum RTD Working Group (2006)
7. Kwon, T., Lee, H., Choi, S., Kim, J., Cho, D.: Design and Implementation of a Simulator Based on a Cross-Layer Protocol between MAC and PHY Layers in a WiBro Compatible IEEE 802.16e OFDMA System. IEEE Communications Magazine 43(12), 136–146 (2005)
8. Jiang, D., Taliwal, V., Meier, A., Holfelder, W.: Design of 5.9 GHz DSRC-based Vehicular Safety Communication. IEEE Wireless Communications 13(5), 36–43 (2006)
9. Kangas, M., Hippi, M., Ruotsalainen, J., Näsman, S., Ruuhela, R., Venäläinen, A., Heikinheimo, M.: The FMI Road Weather Model. HIRLAM Newsletter 51 (2006)

System Architecture for C2C Communications Based on Mobile WiMAX

Michiyo Ashida and Tapio Frantti

VTT Technical Research Centre of Finland,
Kaitoväylä 1, Oulu, P.O. Box 1100, FI-90571, Finland
{michiyo.ashida,tapio.frantti}@vtt.fi

Abstract. In this paper we considered a WiMAX-based system architecture for car to car (C2C) communications. The aim is to design an intelligent wireless traffic service platform which allows car to car and car to transceiver stations communications. Conventional WiMAX system was analysed as a basic platform due to its support for robust security and for mobility. However, we found some problems with the system for supporting C2C communications. As a solution, an optimized C2C communication mechanism with neighbor detection and optimum route decision module was proposed. This module uses available information in the neighborhood and adds no extra cost of traffic.

Keywords: 802.16e, Mobile WiMAX, C2C communications.

1 Introduction

Broadband wireless stands at the confluence of two very remarkable growth stories of the communications industry in recent years. Wireless and broadband have both rapid mass-market adoption. Wireless mobile services grew from 145 million subscribers in 1996 to more than 2.685 billion in 2006 [1]. At the same time, the Internet grew from 74 million users to 1.131 billion. The growth of the Internet is driving demand for higher-speed Internet-access services. During the same period, broadband subscription grew from almost zero to over 200 million subscribers [2]. Therefore, it is plausible that combining the convenience of wireless with the performance of broadband will be the next frontier for growth in the communication industry [3].

The primary aim of the WiMAX (worldwide interoperability for microwave access) system was to find a competitive alternative to traditional wireline-access technologies. It has evolved through four stages, albeit not clearly sequential: narrowband wireless local-loop systems, first generation line-of-sight broadband systems, second generation non-line-of-sight broadband systems, and standards-based broadband wireless systems. The first standard, the original IEEE802.16, completed in December 2001 and amendment for it is know as IEEE 802.16a. Further revisions resulted in a new standard in 2004, IEEE 802.16-2004, which replaced prior to versions and formed the basis for the first WiMAX solution and was targeted fixed applications. In December 2005, the IEEE 802.16e-2005 was

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 558–567, 2008.

completed and approved as an amendment to the IEEE 802.16-2004 standard that added mobility support. It is often referred to as Mobile WiMAX, too [3].

WiMAX is a wireless broadband solution that offers a set of features with a lot of flexibility, such as OFDM physical layer, high peak data rates, scalable bandwidth and data rate support, adaptive modulation and coding, link laayer retransmission, support for TDD (time division duplex) and FDD (frequency division duplex), OFDMA (orthogonal frequency division multiple access), flexible resource allocation, support for advanced antenna techniques, quality of service support, robust security, support for mobility and IP-based architecture [3].

In this paper we consider a system architecture for C2C communication. The aim is to design an intelligent wireless traffic service platform for car to car communication and for car to transceiver stations communication. Transceiver stations are located beside the roads. The primary applications for the system architecture are exchange of real-time local weather data, the urban transport traffic management, and the urban information broadcasting between cars and cars and transceiver stations. The WiMAX system was analysed as a basic platform due to its support for robust security and for mobility. Therefore, the organisation of the rest of the paper is following. In section 2 a literature review from the WiMAX based C2C communications is presented. Section 3 introduces existing system model for C2C communications. Section 4 describes a developed optimized module for C2C communication. Section 5 analyzes developed solutions and compares them to the exisisting solutions. Finally, conclusions are presented in section 6.

2 Related Work

2.1 Reference Network Architecture

The IEEE 802.16e-2005 standard provides the air interface for WiMAX but does not define the end-to-end WiMAX network. The Network Working Group (NWG) of WiMAX Forum is responsible for developing the end-to-end requirements, architecture, protocols and communication mechanims for WiMAX using IEEE 802.16e-2005 as the air interface. The NWG has developed a network reference model to serve as an architecture framework for WiMAX deployments and to ensure interoperability among WiMAX equipment and operators. The reference model envisions a unified network architecture for supporting fixed, nomadic and mobile deployments and it is based on an IP model. The overall network can be logically divided into three parts: mobile stations, an access service network (ASN), and a connectivity service network (CSN). Mobile stations are used by end users to access the network. The ASN comprises one or more base stations and one or more ASN gateways. The CSN provides IP connectivity and all the IP core network functions. [3]

2.2 Mobile WiMAX Car Implementations

In April 2005, KDDI succeeded testing of handover between Mobile WiMAX Base Stations (BSs) and also with 3G, with applications such as multi-channel

streaming, high quality VoIP and media switching [4]. In January 2008 at Consumer Electronics Show (CES 2008), Intel demonstrated Mobile WiMAX car racing show in which real-time video taken by the car was broadcasted to the main exhibition hall. At the same CES 2008, Oki along with Alpine and Runcom demonstrated their navigation system based on Mobile WiMAX. These proved high performance of Mobile WiMAX networks with a variety of broadband applications such as VoIP, video streaming, and large-size file transfer. However, these implementations were based on Point-to-MultiPoint (PMP) mode and had no capability of exchanging messages between cars.

2.3 C2C Communications Based on WiMAX

To the best of our knowledge, no C2C Communication System has been implemented until present. Several research has targeted at integration of routing and scheduling mechanism for WiMAX vehicular networks [5] and Intelligent Transportation System (ITS) [6], and design of C2C communication protocol for fair multihop data delivery [7]. All of these are based on Mesh mode of Mobile WiMAX standard. Mesh architecture is suitable for a network with medium to high density. It expands coverage area and enables C2C communications. In reality, however, it is likely that density of a traffic network vary due to many reasons, for example deployment area and time of the day. Also multihop forwarding may increase management cost, latency and packet loss rates.

2.4 Neighbor Detection Algorithms

HELLO based neighbor detection is used in many ad hoc routing protocols such as AODV [8], DSDV [9] and OLSR [10]. In these algorithms, each node periodically broadcasts HELLO messages and advertises itself to its neighbors. This HELLO message may include information of the transmitter's neighbors, which is useful for establishing multihop routes. Because message exchange is needed more frequently in real-time mobile environment, HELLO based neighbor detection algorithms may reduce data throughputs in the whole network.

Handshake based neighbor detection is often used to avoid hidden terminal problem. A node first send probe packet and the receiver replies with ACK. This type of algorithm can be initiated by the sender whenever it needs to. A well-known example of handshake based neighbor detection algorithm is RTS/CTS (Request to Send / Clear to Send) algorithm adopted by IEEE 802.11. RTS/CTS requires four-way message exchange.

Link state aware neighbor detection takes quality of wireless channel into account. As an option, OLSR support usage of link state information for its neighbor node management system. ETX [11] computes channel quality based on successful delivery of probe packets in a given time frame. Because link state is constantly monitored, link state aware neighbor detection is more reliable than HELLO based or handshake based algorithms in which decision making is based only on few times of message exchange. The disadvantage of this type of algorithm is high cost of periodic probe transmissions.

3 Existing System Model for C2C Communications

In this section, we consider WiMAX based system architecture for C2C communications. We first introduce assumptions and requirements, then point out problems with existing architectures, and finally analyze necessary functionalities to enable C2C communications based on Mobile WiMAX.

3.1 Assumptions and Requirements

Our deployment scenario, a large-scale broadband wireless platform for local road weather services, is mainly targeted at city or highway area. In our platform, cars are driving freely at high speed (up to 100km/h), and constantly collecting data from car sensors while travelling. Collected data from each car is transmitted using our wireless platform and shared with data analysis centers and other nearby cars. Each car is assumed to be equipped with omni-directional antenna with the same transmission power. We also assume TDD duplex method for the system. Table 1 shows system requirements derived from our scenario. Wireless technologies (3G, WiMAX and WiFi) have been evaluated based on these requirements. Scaling is based on three levels: A, B and C, in higher order.

3.2 Operation Modes of Mobile WiMAX

A Mobile WiMAX network consists of two types of devices: BS and Mobile Subscriber Station (MSS). MSSs are registered to and controlled by the BS strictly with node management system, such as device authentication, and resource allocation mechanisms. With Mobile WiMAX, two independent operation modes are defined: PMP and Mesh.

PMP Mode. In PMP mode, every MSS is accessible from/to BS with a single hop. The advantage of PMP mode is its simplicity. It is easy to deploy and simple to manage. For this, implementation of PMP mode is mandatory in every Mobile WiMAX device.

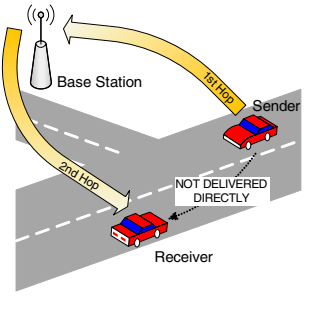


Fig. 1. Packet forwarding problem in PMP mode. Packets are always relayed by the Base Station node, even when the sender and the receiver are in communication range.

Table 1. Comparison of wireless technologies based on our requirements

Requirement		Cellular (3G)	Mobile WiMAX		WiFi	
#	Details		PMP Mode	Mesh Mode	Infrastructure Mode	Ad Hoc Mode
1. Wireless broadband connection between car and Base Station (1km)						
1 (a)	Long distance transmissions (large coverage area)	A	A	A	B (Medium range)	B (Medium range)
1 (b)	Reachability from BS to car and vice versa within a given network, to assure successful delivery	A	A	B (Highly dependent on the topology)	C (Not for long distance)	C (Not for long distance)
1 (c)	Wide bandwidth to achieve high throughput	B (Slower speed than others)	A	A	A	A
1 (d)	Physical-layer support for Non-Line-of-Sight (NLOS) transmissions	A	A	A	A	A
1 (e)	Scalable node management at BS to support a large number of nodes	A	A	A	B (Less capacity)	C (No central management node)
1 (f)	Sophisticated Media Access Control (MAC) support to provide fair radio access opportunity to every nodoe	A	A	A	B (Not efficient for a large network)	B (Not efficient for a large network)
1 (g)	Low user cost for packet transmission	C (Usually, charged based on data size)	A (Public network) or B (Commercial)	A (Public network) or B (Commercial)	A (Public network) or B (Commercial)	A (Public network) or B (Commercial)
1 (h)	Security support	A	A	A	B (Shared secret encryption scheme does not support privacy of individual node)	B (Shared secret encryption scheme does not support privacy of individual node)
2. Car to car communication						
2 (a)	Ability of node to dynamically discover and manage its neighbor nodes	C (No such mechanism)	C (No such mechanism)	A	C (No such mechanism)	C (No dynamic configuration)
2 (b)	Possible to send/receive packets within local network without routing through BS	C (No such mechanism)	C (No such mechanism)	B (May have no compatibility with other nodes due to non-standardized protocols)	C (No such mechanism)	B (May have no compatibility with other nodes due to non-standardized protocols)
2 (c)	Physical-layer support for NLOS transmissions	Refer to 1 (d)				
2 (d)	Security support	Refer to 1 (h)				
3. Real-time data delivery						
3 (a)	Wide bandwidth to achieve high throughput	Refer to 1 (c)				
3 (b)	Small end-to-end latency	B (Slower due to limited bandwidth)	A	B (Slower due to relay over multiple wireless links)	A	B (Slower due to relay over multiple wireless links)
3 (c)	Quality-of-Service (QoS) management	A	A	A	C (No such mechanism)	C (No such mechanism)
4. Improve user mobility						
4 (a)	Capable of having fast moving nodes	A	A	B (Management of fast moving nodes is costly)	C (Nomadic within a network, but no inter-network mobility)	C (Nomadic within a network, but no inter-network mobility)
4 (b)	Long distance transmissions (large coverage area) for better connectivity	Refer to 1 (a)				
4 (c)	MAC-layer support for handover	A	A	A	C (No such mechanism)	C (No such mechanism)

Although many traffic or transport services can increase usability and mobility by using C2C communications, Mobile WiMAX's PMP mode does not support this type of communication. As illustrated in Fig. 1, even when the sender and receiver MSSs are close enough from each other, packets are always routed through the BS and arrive at the destination after two wireless hops.

This problem is caused by lack of knowledge at MSSs about their neighbor nodes. Our study showed that the destination MSS may hear the signals from the source, but cannot receive the packet successfully. Because the sender does not know if the destination is in its transmission range, the only choice for the next-hop MAC address used in the MAC header is the BS's. Unmatching of MAC addresses will be automatically detected and filtered out by hardware at the receiver MSS.

Mesh Mode. In Mesh mode, packets may be exchanged between MSSs or transferred by multiple hops with help of other MSSs. Thus, direct connection to the BS is not necessary for MSSs in this mode. Some of the advantages of Mesh mode are communication within local community and expansion of coverage area with minimum hardware investment.

The Mesh mode is optional and not compatible with PMP mode, which means that Mobile WiMAX Mesh networks can be possible only when all the nodes have implemented this optional mode. In fact, however, the majority of Mobile WiMAX devices in market are only capable of PMP mode.

Achieving sufficient system performance with wireless multihop architectures has been a big research challenge for many years, due to constrains of radio transmissions and complexity of routing and node management. Transmissions over multiple wireless hops may cause degradation of system performance, not only by increasing end-to-end delay but also by adding extra cost of using more bandwidth and more risk of packet losses [12]. Ken Stanwood states following challenges for mesh architectures: delays (transmission delays and processing delays) due to multiple forwarding, limited capacity at tree branches, higher load at root node, poor QoS achievement because of delay, expensive cost of planning and management, complex routing, and compatibility problem with PMP mode due to unique frame structure [13].

Furthermore, network connectivity of Mesh architectures is highly dependent on the topologies. In a network where nodes keep moving freely at high speeds, a fatal error may occur when there is no relaying MSS between the sending MSS and the BS.

Although direct communications between neighbors is possible with Mesh mode, we have concluded that assuring connectivity is the first priority and thus Mesh is not the ideal solution for our wireless transport service platform in which topology keeps changing along with the node movements.

3.3 Enabling C2C Communications for Mobile WiMAX

Our motivation of enabling C2C communications for our wireless transport system is to increase usability and mobility, by optimizing transmission mechanism

and minimizing several costs due to transmissions. Based on the system requirements and analysis explained in 3.1, we have selected Mobile WiMAX as the best candidate technology for our large-scale wireless transport system. Deeper analysis of Mobile WiMAX modes in 3.2 proved that PMP mode was more suitable architecture than Mesh mode because of its wider compatibility, better performance, and reliable connectivity, except that it still needs some mechanisms in order to enable C2C communications. The fundamental mechanisms to enable C2C communications based on Mobile WiMAX are: neighbor detection, neighbor list management and routing decision making. It is our challenge that the costs introduced by these new mechanisms, especially amount of traffic for neighbor detection, should not exceed the original ones with PMP mode.

4 Optimized C2C Communication Mechanism

The future mobile communication platform for vehicles should support a variety of services and allow sharing of localized information between cars and/or car and the backbone infrastructure.

Our optimized C2C communication system is designed to add C2C communication capability to the standard Mobile WiMAX networks with PMP mode. It uses existing traffic for neighbor detection, and select optimum route based on the neighbor information. As a result, this system can achieve shorter packet delivery time and conserve bandwidth for future needs.

Some of the characteristics of the system are:

- As efficient as PMP mode (no extra cost of delay, packet loss, scheduling)
- Direct communication within local community is possible (less load at BS, less bandwidth consumption)

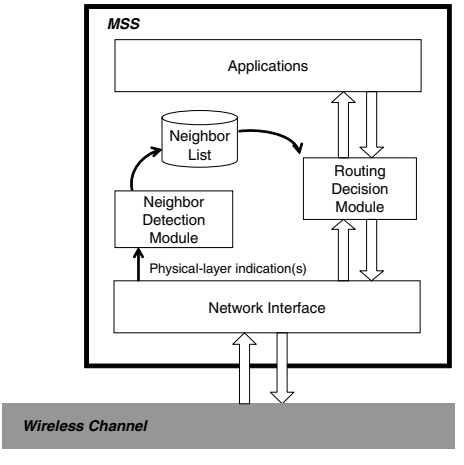


Fig. 2. Node components for Mobile Subscriber Station (MSS)

– Neighbor discovery based on existing packets' physical-layer information
– Adds broadcasting functionality at MSSs. Can be used for more localized broadcasting (MSS's transmission range is usually smaller than that of BS).

As illustrated in Fig. 2, our system consists of three components: neighbor detection module, neighbor list, and routing decision module. We describe each component in the following subsections.

4.1 Neighbor Detection Module

The neighbor detection module in our system senses physical indications of surrounding nodes by listening to the signals.

– Step 1: Listen to incoming signals and obtain physical-layer parameters
– Step 2: Based on the parameters, compute one-hop away neighbors using pre-defined threshold
– Step 3: If a neighbor is detected, update *Neighbor List*

As we have discussed in Section 2, the majority of existing neighbor detection mechanisms detects neighbors by sending some message(s). For such ad hoc networks, ensuring connections with neighbors is important since there is no alternative path.

In our case with PMP mode, we already have a default route (connection) to the BS, and want to optimize the route when direct connection to the receiver is found possible. Thus, we avoid costly message-sending neighbor detection approach but aim at using already available information in the neighborhood. Physical information based detection approach can be seen in WiFi networks, when a client selects the best Access Point to connect. Our solution adopt the idea from WiFi's AP detection mechanism, but apply it for neighbor MSS detection by using available packets.

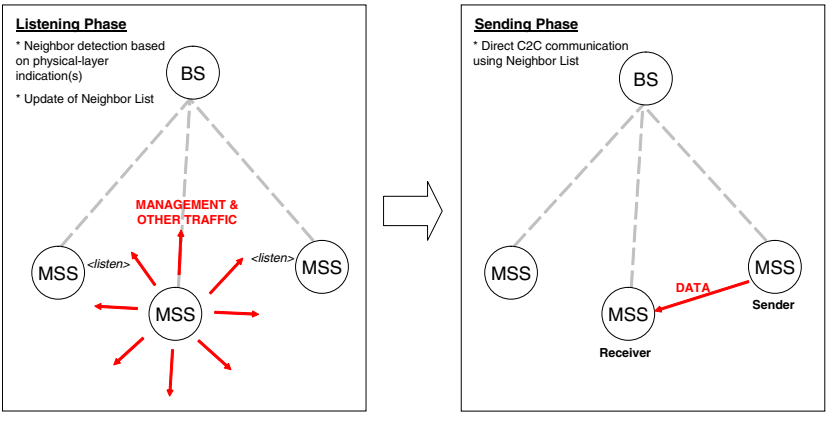


Fig. 3. Neighbor detection and C2C communication processes

Obtaining information from existing packets is rather easy in Mobile WiMAX networks. Firstly, because each MSS is periodically and frequently communicating with the BS, for example to send resource allocation request or some other control messages. Secondly, it is possible for a MSS to catch uplink traffic from other MSSs, because the uplink and the downlink traffic use the same channel (frequency) in TDD.

4.2 Neighbor List

Neighbor List is used to manage all neighbor nodes' information. It simply keeps track of information such as IP address and expiration time for each neighbor entry. Because the information is used for real-time routing decision module, each entry expires after a certain period of time if it does not hear any new information from the same node.

4.3 Routing Decision Module

Routing decision is made at Routing Decision Module using *Neighbor List*. The choice is either one direct hop to the receiver if the node is in *Neighbor List*, or otherwise one hop to the default gateway (the BS).

After routing decision is made, packet header is prepared accordingly. Once the receiver is detected as its neighbor, packets will be automatically directed to the receiver node, and they no longer need to have the BS as intermediate forwarder.

5 Discussion

In a conventional Mobile WiMAX network with PMP mode, communication amoung MSSs on the same IP subnet was not possible at all because packets were always relayed by the BS. Our optimized C2C module provides added functionality for C2C communications to Mobile WiMAX networks in PMP mode. By optimizing the routing decision, it reduces end-to-end latency and packet loss rates. It also conserves bandwidth for future use. Because this module uses available information in the neighborhood for neighbor detection, no extra traffic is produced.

However, misuse of routing metric may cause more packet drops and degrade system performance. In the future research, we will simulate our system and conduct experiments to find reliable trigger point for routing decision making.

6 Conclusion

In this paper we considered a WiMAX-based system architecture for C2C communications. The WiMAX system was analysed as a basic platform. We found that the WiMAX was not an ideal system for C2C communications due to lack of mechanisms, such as neighbor detection, neighbor list management and routing

decision modules. We have designed these functionalities in our optimized C2C module, with aim of conserving radio resource at the same time. Significance of this module is that it uses existing traffic for neighbor detection. The neighbor detection module senses physical indications of surrounding nodes. Based on this neighbor information, optimum route is selected for transmissions. Our optimized C2C communication mechanism can be used in Mobile WiMAX networks with PMP mode, and it reduces extra cost of multihop forwarding, improves end-to-end latency and decreases bandwidth usage.

References

1. ITU. Telecommunications indicators update-2006. International Telecommunication Union (2006)
2. Paxton, M.: The Broadband Boom Continues: Worldwide Subscribers Pass 200 Million. IN0603199MBS (March 2006)
3. Andrews, J.G., Ghosh, A., Muhamed, R.: Fundamentals of WiMAX: Understanding Broadband Wireless Networking. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice Hall PTR, Englewood Cliffs (2007)
4. KDDI. KDDI and WiMAX - Convergence in the Land of the Rising Sun. White Paper. WiMAX Forum (August 2006)
5. Amin, R., Wang, K., Ramanathan, P.: An Integrated Routing and Scheduling Approach for Persistent Vehicle Communication in Mobile WiMAX Mesh Networks. IEEE (2007)
6. Chang, B., Huang, B., Liang, Y.: Wireless Sensor Network-based Adaptive Vehicle Navigation in Multihop-Relay WiMAX Networks. In: 22nd International Conference on Advanced Information Networking and Applications. IEEE Computer Society, Los Alamitos (2008)
7. Yang, K., Ou, S., Chen, H., He, J.: A Multihop Peer-Communication Protocol With Fairness Guarantee for IEEE 802. 16-Based Vehicular Networks. IEEE Transactions on Vehicular Technology 56(6) (November 2007)
8. Perkins, C.E., Belding-Royer, E.M., Das, S.R.: Ad Hoc On-demand Distance Vector (AODV) Routing, IETF RFC 3561 (2003)
9. Perkins, C.E., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. SIGCOMM 1994 (1994)
10. Clausen, T., Jacquet, P., et al.: Optimized Link State Routing Protocol, IEEE INMIC Pakistan (2001)
11. De Couto, D., Aguayo, D., et al.: A High Throughput Path Metric for Multi-Hop Wireless Routing. ACM MobiCom (2003)
12. Biswas, S., Morris, R.: Opportunistic Routing in Multi-hop Wireless Networks. In: Proceedings of IEEE/ACM SIGCOMM (2005)
13. Stanwood, K.: WiMAX and Mesh Networking in the Home. WiMAX and Mesh Networks Forum, The IEE, Ref. No. 2005/11919 (2005)

Accuracy and Efficiency in Simulating VANETs

Enrique Alba, Sebastián Luna, and Jamal Toutouh

Dept. de Lenguajes y Ciencias de la Computación, University of Málaga,
ETSI Informática, Campus de Teatinos, Málaga - 29071, Spain
{eat,sebastian,toutouh}@lcc.uma.es

Abstract. The evaluation of new communication protocols for *Vehicular Ad-hoc Networks (VANETs)* is a hot topic in research. An efficient design and actual deployment of such software tools is crucial for any VANET. The design phase is difficult and often relies on computer simulation. The later evaluation of protocols in real VANETs is complex due to many difficulties concerning the availability of resources, accurate performance analysis, and reproducible results. Simulation is the most widely solution to make a good design but it presents also an important challenge: the fidelity of the simulation compared to the real results. In this article we measure the differences between the simulation versus the real results with actual moving cars in order to quantify the accuracy of the VANET simulations inside the *European CARLINK Project*[1]. After a thorough revision of the state of the art, we here go for an analysis of JANE and VanetMobiSim/ns-2, two simulation frameworks. Later, we have defined the scenario where both, simulations and real tests, will be carried out. Our results show that JANE is more appropriate for simulating applications, while ns-2 is more accurate in dealing with the underlying mobile communication network.

Keywords: CARLINK, IEEE 802.11b, Simulation, VANETs, JANE, VanetMobiSim/ns-2.

1 Introduction

Vehicular Ad-hoc Networks (VANETs) are created by equipping vehicles with devices capable of wireless communication. The existence of such networks opens the way for a large range of applications: providing real-time information about traffic jams, accidents, and weather; that could be useful for developing a wider set of car vehicles that keeps people connected in metropolitan routes and highways in a clear advance to safer driving, a main issue in today's society.

The evaluation of VANET protocols and applications in outdoor experiments, on using large-scale networks to obtain significant results, is extremely difficult for reasons like the limited/dynamic set of available resources, inaccurate performance analysis, and often irreproducible results. Indeed, it is neither easy nor cheap to have a high number of real vehicles and a real scenario amenable for

[1] http://carlink.lcc.uma.es

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 568–578, 2008.

VANET designers. It is also difficult to analyze protocols performance in a inherently distributed, changing, and complex environment like a VANET [1].

Hence, simulation has become an indispensable tool. It allows to *build* inside a computer a dedicated VANET for the evaluation of protocols: the number of vehicles, the direction and velocity of their movement, the features of the wireless network transceivers, the routing protocol, etc. Simulators also gather statistical data about the network usage during the simulation, which allows to measure the protocols performance. Moreover, it is possible to visualize the VANET in order to easily analyze and conclude on protocol evaluation.

However, due to the complexity of any real scenario in which cars move, a big amount of information related to the signal propagation is missed what is bad news because it plays an important role in the performance of the outdoor experiments: passing by obstacles, reflection problems, coverage signal interferences, etc. Thus, simulation also presents an important drawback: the fidelity of the generated results.

The aim of the CARLINK Project is to develop an intelligent wireless platform among cars. The global scenario considers the cars as data collectors that sends relevant information via wireless technologies up to a central station (where this info is processed). Inside this global scenario, the ad-hoc communications allows the cars to communicate directly with each other without the need of existing infrastructure.

In this paper, we first focus on the simulation of the *CARLINK-UMA scenario*. This scenario consists in two cars moving at 30 km/h while transferring different files between them. The cars follow different mobility models and they are connected through the ad-hoc operation mode of the IEEE 802.11b standard. The goal is to also reproduce the same scenario in outdoor experiments in order to compare the simulated data with the real ones.

We have found that the results of the real tests performed at the University of Málaga (UMA) are an accurate estimation of the data rates that can be achieved when using the ad-hoc WiFi for transferring files between cars. These results could determine the type of applications that could run on top of VANETs. To this goal, we use several simulators, we consider communications and applications, and quantitatively analyze the results.

This paper is structured as follows: Section 2 summarizes the different alternatives for VANETs simulation. Later, VanetMobiSim/ns-2 and JANE are selected as the simulators of interest for the CARLINK Project and Section 3 gives an overview about them. Section 4 defines the scenario where both, the simulations and the real tests, will be carried out. Section 5 presents the simulated versus the real experiments. Afterwards, these results are compared with each other, and finally, Section 6 draws some conclusions about the methodology as well as on the achieved results.

2 Simulation of VANETs

VANETs are a subclass of the Mobile Ad-hoc Networks (MANETs) in which mobility patterns are more complex, since the network topology changes more

frequently because of the higher node velocity and the nodes having to fulfil the traffic rules. Therefore, a realistic mobility model implementation is as relevant as a realistic ad-hoc communication network model in order to obtain good quality VANET simulation results. Let us first discuss appropriate tools for simulating either communications and mobility models in next section.

2.1 VANET Simulation Alternatives

Nowadays, we identify different approaches trying to through light on the complex problem of simulating VANETs in a trustworthy manner. First, the most widely used, the desingner could use a traffic simulator for generating realistic vehicular mobility traces that will be used as the input for a mobile ad-hoc network simulator. Second, the designer could use a specially-designed VANET simulator tool. Finally, some MANET application programming frameworks allows the developer to test the applications via simulations.

The first approach used for simulating VANETs consist on using a traffic simulator or a mobility model generator capable of generating mobility traces, which are later evaluated by an existing specific MANET simulator. The public availability of many of these MANET simulators is the main motivation for the success of this approach. However, it has a major drawback: the majority of VANET applications need vehicles to react to network events and it is difficult to be modeled with this scheme of simulation. Most research community adopt ns-2 (network simulator) [8] for MANET simulating. The number of traffic simulators which generates ns-2 format traces is large: the most comprehensive is **VanetMobiSim** [3], however we can also find another as Videlio, RoadSim, CARISMA, VISSIM, and MMTS. There are also traffic simulators that generate traces for other MANET simulators as CORSIM/TSIS, SJ04, SSM/TSM, and STRAW. Finally, TraNS and MOVE combine the SUMO mobility model generator and ns-2 simulator linking them in a unique tool.

The specially-designed VANET simulators join scalable vehicular mobility descriptions and network stack modelling in a single tool. These combined approaches have the big advantage of allowing a direct interaction between the communication network system and the vehicular traffic dynamics, thus, the first can influence the second. However, they also have a major drawback. The level of detail of both modules is necessarily lower than that provided by ad-hoc simulation tools. GGCDCI06, MoVES, and the GrooveNet are examples of specific VANET simulators.

Finally, there are some frameworks as JANE [4], a Java-based middleware platform for MANET applications programming. It allows the developer to test the applications in a simulation environment and, also, over real mobile devices. See [5] for more details.

2.2 Selecting a Simulator

Once revised the different approaches for VANET simulation, this section is devoted to giving some recommendations for choosing the most appropriate

tool. First of all, the software which is distributed under commercial licenses, as most specific VANET simulators, constitutes a major flaw to adopt them by the research community. Thus, the use of a traffic simulator that consists of the traffic generator traces and the MANET simulator is the most suitable choice. We decided to use ns-2 as MANET simulator, since it is widely used by the research community. So, we need a simulator to generate ns-2 format traces.

Furthermore, the selected simulator has to generate realistic mobility models that reflects as closely as possible the characteristic behavior of the nodes as real vehicles through road traffic by using *macro-mobility* and *micro-mobility* definitions [2]. The simulator should be intuitive with no complex mobility definition. This leaded us to also experiment with JANE [4], in order to have simulations coupled to applications one of its salient feature.

3 VanetMobiSim/ns-2 and JANE Simulators

According to the previous recommendations, the chosen traffic simulator is VanetMobiSim. It includes several options to specify the roads characteristics (macro-mobility features) and the behavior of the mobility of the nodes (micro-mobility features), the definition of them is done by using intuitive XML code, and the output trace has ns-2 format. Moreover, JANE has been also selected since it allows the development, simulation, and execution of high-level applications in an integrated way.

3.1 VanetMobiSim/ns-2 Simulator

The simulator used for most of the simulations in CARLINK is the combination of the traffic simulator VanetMobiSim and the MANET simulator ns-2 [7].

VanetMobiSim is an extension to CanuMobiSim [9], a generic user mobility simulator. CanuMobiSim provides an efficient and easily extensible mobility architecture, but due to its general purpose nature, it suffers from a reduced level of detail in specific scenarios. VanetMobiSim is therefore aimed at extending the vehicular mobility support of CanuMobiSim to a higher degree of realism. The main characteristics of this simulator are that it is specific for VANETs and an open source platform; it supports both macro-mobility and micro-mobility specification, and it uses intuitive XML code to specify the different simulations. However the most important feature of VanetMobiSim is that it has been validated in actual communication scenarios [3] . Its main drawback is that it offers a poor documentation.

ns-2 [8] is an open source network simulator, so it is freely available and the user is able to modify the source code (C++ and OTcl). This characteristic is really important, since it has allowed us to extend the simulator with the VDTP protocol [6]. It provides a packet level simulation over a lot of protocols, supporting several transport protocols, several forms of multicast, wired networking, several ad-hoc routing protocols and propagation models, data broadcasting, satellite, etc. It incorporates different traffic generators as web, telnet,

CBR (constant bit rate generator), etc. for using them in the simulations. Also, ns-2 has the possibility of using mobile nodes. The mobility of these nodes may be specified either directly in the simulation file or by using a mobility trace file. In our case, the trace file is generated by VanetMobiSim. Finally, other important feature is that it incorporates several add-ons as the visualization tools NAM[2] (Network Animator) and TraceGraph[3].

3.2 JANE: The Java Ad-Hoc Network Environment

JANE [4] is an open source Java-based middleware platform which is intended to assist ad-hoc network researchers in application and protocol design. JANE aims at providing a uniform workbench, supporting experiments ranging from pure simulation of mobile devices, over hybrid scenarios with interaction among simulated as well as real life devices, up to dedicated field trials as proof of concepts. Therefore, JANE presents three different execution modes that enable the execution of the tested source code from the simulation to the real devices with a low effort. These execution modes are: *simulation mode*, *hybrid mode*, and *platform mode*. In simulation mode, the complete environment is simulated: the devices, the users and the ad-hoc network. In hybrid mode, the devices and the ad,hoc network are simulated, but real users can interact with the simulation by using emulated graphical interfaces. Finally, in the platform mode the whole setting is real (actual mobile devices as PDAs, cellular phones, etc.).

A development process can be derived from the utilization of these three JANE execution modes. It consists of a cycle which comprises three ordered phases: simulation, emulation, and real execution. It starts with implementing, testing, and evaluating algorithms and applications in a purely simulated environment. In a second step, dedicated mobile devices can be cut out of a simulation run and be transferred to a real mobile device in order to deal with real user interaction and to evaluate the user experience. In a final phase, specific field trials can be defined and executed on real mobile devices.

The main disadvantage of JANE is that this tool is not specialized for VANETs. Therefore, it does not provide realistic mobility models for the simulation of vehicular networks. Nevertheless, its well structured simulation kernel could allow the developer to integrate a more accurate mobility model component for overcoming this weakness.

4 The CARLINK-UMA Scenario

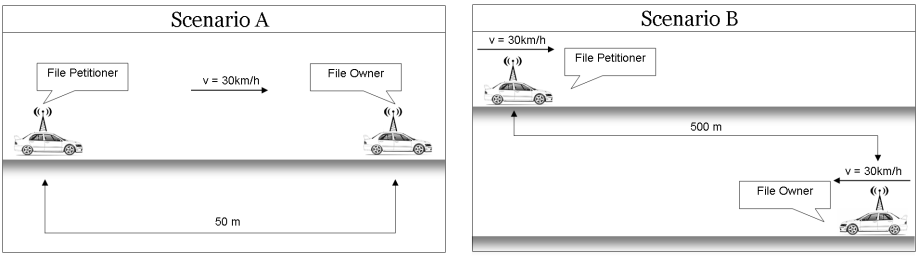
The CARLINK Project consider three scenarios for the exploitation of the intelligent wireless platform that is going to be built at the end of the Project [10]: *the local weather service*, *the traffic management service*, and *the mobile end user service*. These services use both, infrastructure and ad-hoc communications, to

[2] http://www.isi.edu/nsnam/nam
[3] http://www.angelfire.com/al4/esorkor

achieve their objectives. In the CARLINK-UMA scenario we focus on the study of the inter-vehicle communications quality in which the VANETs are based on.

This section describes the scenario where the simulations and the real experiments will be carried out. These conditions will be exactly the same in all the cases in order to make a fair comparison. The goal is to transfer files between two cars connected by using the ad-hoc operation mode of the *IEEE 802.11b MAC Layer Standard* in a line-of-sight scenario. Concretely, each car is equipped with one *PROXIM ORiNOCO PCMCIA transceiver*[4] connected to a range extender antenna. The wireless network cards output power is 12 dBm and the range extender antennas gain is 7 dBi.

The mobility model consists of a road segment split into two lanes representing bi-directional traffic. Depending on the initial and final positions, we differentiate two scenarios: **Scenario A** and **Scenario B** (see Figure 1). In the first one, both vehicles start at the initial position of the same lane, and they move in the same sense along this lane separated by average 50 m (Figure 1.a). In Scenario B, one vehicle starts the movement at the initial position of the first lane and the other vehicle starts at the final position of the second lane, 500 m separated one from the other, and they move in opposite directions (Figure 1.b). In both cases, each vehicle move with a velocity equal to 30 km/h on average.



a) Scenario A b) Scenario B

Fig. 1. Mobility models

The experiments were composed of different tests. Each one consisted in continuously transferring a data file in one of the previously specified scenario A or B (Figure 1). We used two different files: **file 1** with 1-MB size (representing traffic notifications, e.g. on road conditions) and **file 2** with 10-MB size (representing multimedia files, e.g. podcasting or streaming to cars).

We use the VDTP protocol [6] to make transfers among the vehicles. For each transfer, VDTP splits the file into several chunks. The chunk size can be configured manually. We have set its value to 25 KB in all the tests.

The complete experiment consisted of **ten repetitions** of every test. The tests were named as follows: **Test A1**, **Test A2**, **Test B1** and **Test B2**. In this notation, the upper case character describes the scenario and the number denotes the file used in each test.

[4] `http://www.proxim.com`

5 Real Tests Versus Simulation

This section presents the differences between real and simulated results. Firstly, we present the results of real tests and simulations for each test. Secondly, we present the difference between each simulator and the real tests.

Figure 2 shows the results of transferring ten times the file type 1 in the Scenario A. The mean transmission time in the real tests is 1.618 seconds, with a mean transmission rate equal to 626.992 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 1.679 seconds, with an mean data rate equal to 609.778 KB/s. In the case of the JANE simulations the mean transmission time is 1.8 seconds, with a mean data rate equal to 563.812 KB/s. We can notice the high precision of the simulation with ns-2 compared to the actual values showed by the moving cars.

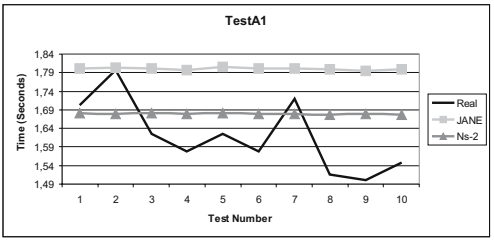


Fig. 2. Individual transmission times for TestA1

Figure 3 shows the results of transferring ten times the file type 2 in the Scenario A. The mean transmission time in the real tests is 17.328 seconds, with a mean data rate equal to 585.176 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 16.757 seconds, with a mean data rate equal to 611.053 KB/s, somewhat too optimistic this time. In the case of the JANE simulations, the mean transmission time is 17.9 seconds, with a mean data rate equal to 564.494 KB/s, a better result for this high level tool.

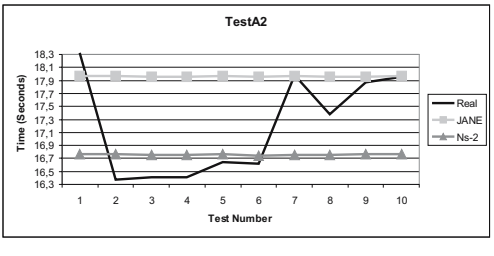


Fig. 3. Individual transmission times for TestA2

Figure 4 shows the results of transferring ten times the file type 1 in the Scenario B. The mean transmission time in the real tests is 2.732 seconds,

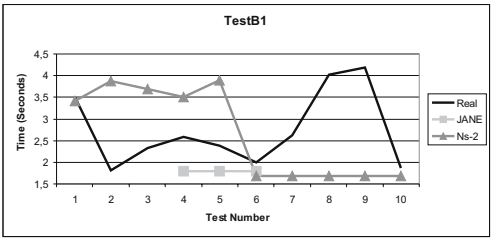


Fig. 4. Individual transmission times for TestB1

with a mean data rate equal to 371.404 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 2.678 seconds, with an average transmission rate equal to 391.451 KB/s. In the case of the JANE simulations the mean transmission time is 1.8 seconds, with a mean data rate equal to 563.724 KB/s. It seems that small data files get JANE more confused, while ns-2 is specially accurate.

Figure 5 shows the results of transferring ten times the file type 2 in the Scenario B. The mean transmission time in the real tests is 20.198 seconds, with a mean data rate equal to 502.017 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 19.945 seconds, with a mean data rate equal to 513.397 KB/s. In the case of the JANE simulations, none of these transfers were successful (i.e., none of the files were completely downloaded from the file owner to the file petitioner).

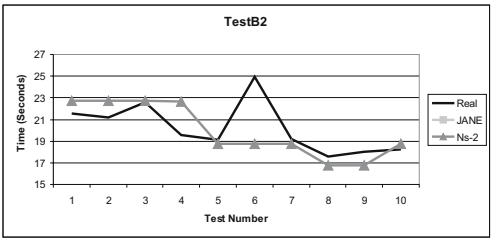


Fig. 5. Individual transmission times for TestB2

In order to compare all these results, Figure 6 presents the mean data rate for each test. It is easy to check that VanetMobiSim/ns-2 generates more realistic results than JANE. Anyway, let us have a look to the numerical differences presented in Table 1. Each entry (i, j) in this table denotes the absolute difference (in KB/s) between the real experiment and the simulation results with the simulator i in the test j.

ns-2 presents the largest difference with the real experiment in Test A2: 25.877 KB/s. JANE presents the largest difference with the real experiment in Test B1: 192.32 KB/s. Moreover, it was not possible to transfer any file completely in JANE with the same conditions as VanetMobisim/ns-2.

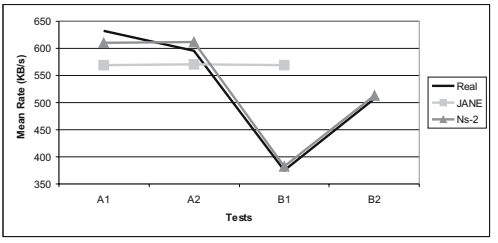


Fig. 6. Mean data rate comparison between real and simulated tests

The configuration of the wireless network transceivers for each vehicle were copied from the real specifications of the ORiNOCO card to both simulators (detailed in [4]). However, the coverage radius for the cars had very different values in each one: 80 metres in JANE and 100 metres in ns-2. The smaller the coverage diameter the smaller the time frame for the connection between the file petitioner and the file owner in the Scenario B. That could be the reason for the observed differences and advantages of ns-2 since it has a larger coverage. Furthermore, the mean data rate achieved in JANE was also smaller than the one achieved in ns-2 during all the tests. This explains the difficulties for JANE in order to transfer the 10-MB file in the Test B2.

Table 1. Mean data rate differences (absolute value in KB/s) between real and simulation results

	Test A1	Test A2	Test B1	Test B2
JANE	63.18	20.682	192.32	N/A
VanetMobiSim/ns-2	17.214	25.877	20.04	11.38

6 Conclusions

In this work we have compared simulated versus real experiments about the use of ad-hoc WiFi in VANETs. Firstly, we have deeply study the state of the art in VANET simulation in order to select the most interesting tools for the CARLINK Project: JANE and VanetMobiSim/ns-2. Secondly, we have defined a common scenario for the fair comparison between simulation and real results: the CARLINK-UMA scenario. Finally, we have presented the numerical differences among them.

It is interesting to notice that the times between consecutive file transfers in the simulations are very similar each other, contrary to the times obtained in the real experiments (see figures 2, 3, 4 and 5). The simulation experience shows that the real world communications are quite difficult to simulate in a trustworthy manner. Due to its complexity, a lot of events related to the signal propagation of the wireless transceivers, that play an important role in the real experiments, are missed in the simulations: passing by obstacles, reflection problems, signal

interferences, etc. It is advisable to keep this idea in mind when using the simulation results to evaluate any complex scenario before being deployed.

The results presented in Section 5 reveal that VanetMobiSim/ns-2 is the most realistic VANET simulator. Therefore, we have decided to use it in order to perform further complex and larger-scale simulations for the CARLINK consortium. Finally, due to its innovative method for developing new wireless ad-hoc network applications, JANE is useful for testing complex high-level applications deployed on VANETs. Indeed, we have developed two applications that have been successfully tested in real VANETs: FSF and Puzzle-Bubble. These applications are available for download from the CARLINK Project web site[5].

The results achieved with VanetMobiSim/ns-2 are similar enough to the ones obtained in the real experiments to consider this simulator as a reliable alternative for the evaluation of the communication protocols for CARLINK.

As a future work we plan to perform more complex simulations. Once we have tuned the simulators configuration, we are able to simulate more realistic scenarios in order to predict the performance of the real ones, e.g. urban and highway environments.

The aim of the CARLINK Project is to develop an intelligent wireless platform among cars that will provide three main services to improve the day to day life of European drivers and citizens: *the traffic local weather service* will offer accurate local weather forecast, *the traffic management service* will afford real time traffic information for drivers and *the mobile end user service* gives useful information to the citizens in order to choose the better route to reach their destination, through private or public transportation systems. All these services are supported by wireless communications by means of infrastructure and ad-hoc communications. Therefore, the quality of the inter-vehicular communications in VANETs is crucial for the success of the platform.

Acknowledgments. This work has been partially funded by several institutions: the Spanish Ministry of Industry under contracts FIT-330210-2006-49 and FIT-330225-2007-1 (CARLINK), the Spanish Ministry of Education and Science under contract TIN2005-08818-C04-01 (OPLINK), and the Regional Government of Andalusia under contract P07-TIC-03044 (DIRICOM).

References

1. Fiore, M., Haerri, F.F., Bonnet, C.: Understanding vehicular mobility in network simulation. In: MoVeNet 2007. 1st IEEE international Workshop on Mobile Vehicular Networks, October (2007)
2. Härri, J., Filali, F., Bonnet, C.: Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy. Technical report, Institut Eurécom (2007)
3. Härri, J., Filali, F., Bonnet, C., Fiore, M.: VanetMobiSim: Generating Realistic Mobility Patterns for VANETs. In: VANET: 2006: Proceedings of the 3rd international workshop on Vehicular ad-hoc networks, Institut Eurécom, France, pp. 96–97. ACM Press, New York (2006)

[5] http://carlink.lcc.uma.es

4. Gorgen, D., Frey, H., Hiedels, C.: JANE-The Java Ad Hoc Network Development Environment. In: 40th Annual Simulation Symposium (2007)
5. CARLINK: UMA.: D2006/6-Evaluating VANET Simulators for CARLINK Primary Applications. Technical report, University of Málaga (2006)
6. CARLINK: UMA.: D2006/10-VDTP: A File Transfer Protocol for Vehicular Ad hoc Network. Technical report, University of Málaga (2006)
7. CARLINK: UMA.: D1.3.2-VanetMobiSim/Ns-2: A VANET simulator for CARLINK. Technical report, University of Málaga (2007)
8. The Network Simulator, http://www.isi.edu/nsnam/ns
9. CANU Project, http://canu.informatic.uni-stuttgart.de
10. CARLINK Consortium.: Definition of Scenarios. Technical report (2007)

Design of Highly Nonlinear Balanced Boolean Functions Using an Hybridation of DCA and Simulated Annealing Algorithm

Sarra Bouallagui[1], Hoai An Le Thi [2], and Tao Pham Dinh[1]

[1] Laboratory of Mathematics (LMI), National Institute for Applied Sciences
BP 08, Place Emile Blondel
76131 Mont Saint Aignan Cedex, France
[2] Laboratory of Theoretical and Applied Computer Science (LITA)
UFR MIM, Paul Verlaine University of Metz
Ile de Saulcy, 57045 Metz, France

Abstract. The aim of the research presented in this paper is finding highly nonlinear balanced Boolean functions. These functions are useful for bloc ciphers based on S-boxes. An hybridation of a DC (Difference of Convex functions) programming approach and a Simulated Annealing (SA) algorithm is developed.

Keywords: Boolean function, nonlinearity, balance, DC programming, DCA (DC Algorithm), SA.

1 Introduction

Substitution boxes, aka S-Boxes, are a key component of modern crypto-systems. Several studies and developments were carried out on the problem of building high-quality S-boxes in the last few years. Qualities of such boxes, such as nonlinearity and balance, steer the robustness of modern block ciphers. Designing suitable S-boxes is a difficult task, the objective being to optimize a maximum number of criteria. We propose and compare in this work different approaches for generating such Boolean functions. These functions feature high quality cryptography criteria in order to become good candidates for building high-quality S-boxes. In our work, nonlinearity and balance are the main criteria considered for building high quality S-boxes based on Boolean functions. These properties have been widely studied in the literature (see e.g. [1] [13] and references therein).

Many techniques have been suggested for building highly nonlinear balanced Boolean functions. They vary from random search, hill-climbing, genetic algorithms and hybrid approaches. (see e.g. [1], [3], [11], [12] and references therein). However, due to the very large dimension of this problem in practice, the standard methods in combinatorial optimization such as branch and bound, branch and cut, cutting plan can not be applied. That is why, in a long time there is no deterministic models and methods for it. Very recently the first deterministic approach has been developed in [9] which is based on DC (Difference of Convex functions) programming and DCA (DC optimization Algorithms). The

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 579–588, 2008.

authors reformulated the problem as a polyhedral DC program by using exact penalty techniques in DC programming, and then used DCA, a robust approach in continuous optimization for solving the resulting DC program. Since DCA is a local approach, how to compute good initial points for it is an important question from numerical points of views. In [10] several versions of the combined DCA-GA (Genetic Algorithm) have been proposed.

Exploiting simultanously the efficiency of DCA and SA (Simulated Annealing), in this paper, we will present some combined DCA-SA versions for this problem.

The paper is structured as follows: in Section 2 we present basic notations of the Boolean functions and its cryptographic properties. Section 3 deals with the optimization formulation of the problem and the DCA procedure. Section 4 describes the hybridation DCA-SA approach and the numerical results.

2 Basic Notations

A *Boolean function*: $f : \{0,1\}^n \to \{0,1\}$ is a function which produces a Boolean result.

The *binary truth table* of a Boolean function of n variables, denoted $f(x)$, is the truth table that contains 2^n elements corresponding to all possible combinations of the n binary inputs.

For a given table $x = (x_1, x_2, ..., x_n)$, the Boolean function f can be determined by the last column of its binary truth table, namely a binary vector in dimension 2^n.

Denote by $B := \{0,1\}$. In this work we consider a Boolean function f as a vector in B^{2^n}. Hence the set of Boolean functions, denoted by F is exactly the set B^{2^n}.

The *polarity truth table* of a Boolean function denoted $\widehat{f}$ is defined by $\widehat{f}(x) = (-1)^{f(x)} = 1 - 2f(x)$, where $\widehat{f}(x) \in \{1, -1\}$.

A *linear Boolean function* $L_w(x)$, selected by $w \in Z_2^n$, is a Boolean function given by ($\oplus$ denotes the Boolean operation 'XOR')

$$L_w(x) = wx = w_1x_1 \oplus w_2x_2 \oplus ...w_nx_n. \tag{1}$$

An *affine Boolean function* $A_w(x)$ is a Boolean function which can be represented in the form

$$A_w(x) = wx \oplus c \text{ where } c \in Z_2. \tag{2}$$

Two fundamental properties of Boolean functions are Hamming weight and Hamming distance.

- The *Hamming weight* of a Boolean function is the number of ones in the binary truth table.
- The Hamming distance between two Boolean functions is the number of positions for which their truth tables differ.

Property 1. a) The Hamming weight of a Boolean function is given by:

$$hwt(f) := \sum_{x \in B^n} f(x) = \frac{1}{2}\left(2^n - \sum_{x \in B^n} \widehat{f}(x)\right). \tag{3}$$

b) The Hamming distance between two Boolean functions is computed as

$$d(f,g) := \sum_{x\in B^n} f(x) \oplus g(x) := (2^n - \sum_{x\in B^n} \widehat{f}(x)\widehat{g}(x)). \tag{4}$$

The *Walsh-Hadamard Transform* (WHT) of a Boolean function is defined as: $\widehat{F}(w) := \sum_x \widehat{f}(x)\widehat{L}_w(x)$.

Cryptographic properties

Boolean functions used in cryptographic applications have to satisfy various cryptographic criteria. Although the choice of the criteria depends on the cryptosystem in which they are used, there are some properties (balance, nonlinearity, high algebraic degree, correlation immunity, propagation criteria) which a cryptographically strong Boolean function ought to have. In this paper we will focus on high nonlinearity and balance.

1. **Balance:** A Boolean function is balanced if its output is equally distributed, its weight is ,then, equal to 2^{n-1}.
2. **Nonlinearity:** The nonlinearity of a Boolean function is defined as the minimum Hamming distance of the function from the nearest affine Boolean function.

Property 2. The nonlinearity of a Boolean function f, denoted N_f, is related to the maximum magnitude of WHT values, and given by

$$N_f := 2^{n-1} - \frac{1}{2} \max_{w\in B^n} |\widehat{F}(w)|. \tag{5}$$

3 A Deterministic Optimization Approach: DCA

3.1 A Deterministic Combinatorial Optimization Formulation

The object of our work is to generate highly nonlinear balanced Boolean functions. In other words, we have to find a balanced Boolean function featuring a maximal nonlinearity criterion. According to the above notations and properties, the problem of maximizing the nonlinearity of a Boolean function can be written as ([9]):

$$\max_{f\in F} N_f = \max_{f\in F} \min_{w\in B^n} \frac{1}{2}\left(2^n - |\sum_{x\in B^n} \widehat{f}(x)\widehat{L}_w(x)|\right).$$

Denote by $a_{wx} := \widehat{L}_w(x) \in \{-1, 1\}$ for w, $x \in B^n$. It has been proved in [9] that

$$\max_{f\in F} N_f = 2^{n-1} - \min_{u\in B^n} \Psi(f),$$

where

$$\Psi(f) := \max_{w \in B^n} \left| \sum_{x \in B^n} a_{wx} f_x - \frac{1}{2} \sum_{x \in B^n} a_{wx} \right|.$$

Hence maximizing N_f amounts to minimizing $\Psi(f)$ on B^n. For finding a balanced Boolean function the next constraint is imposed

$$| \sum_{x \in B^n} f_x - 2^{n-1} | \leq b, \tag{6}$$

with a nonnegative number b. Clearly that if $b = 0$, then the function is balanced. Finally the following optimization problem is formulated in [9]:

$$\beta := \min\{\Psi(f) : 2^{n-1} - b \leq \sum_{x \in B^n} f_x \leq 2^{n-1} + b,\ f \in B^n\}. \tag{7}$$

It is easy to see that the function Ψ is a polyhedral convex (by definition, a function is polyhedral if it is a pointwise supremum of a finite collection of affine functions). We are then facing the minimization of a convex polyhedral function with binary variables under linear constraints. It is known that the last problem is in fact equivalent to a *mixed zero-one linear program* (with exactly one continuous variable). In our convenient for DC programming approach, for the moment, we consider the problem in the form (7). This problem is of a very large dimension : 2^n variables and 2^{n+1} constraints.

3.2 Continuous Optimization Formulation

In [9] Problem (7) is reformulated in the form of a continuous optimization problem. Let

$$p : \mathbb{R}^{2^n} \to \mathbb{R} \text{ be the function defined by } p(f) := \sum_{x \in B^n} min\{f_x, 1 - f_x\}.$$

It is clear that p is a nonnegative concave function on $[0,1]^n$. Moreover $p(f) = 0$ iff $f \in B^n$. Hence Problem (7) can be expressed as

$$\min \left\{ \Psi(f) : 2^{n-1} - b \leq \sum_{x \in B^n} f_x \leq 2^{n-1} + b,\ p(f) \leq 0 \right\}. \tag{8}$$

Using exact penalty techniques one gets the more tractable continuous optimization problem ($t > 0$ is the penalty parameter):

$$(Q) \quad \beta = \min \{\Psi(f) + tp(f) :\ f \in K\},$$

where

$$K := \left\{ f : 2^{n-1} - b \leq \sum_{x \in B^n} f_x \leq 2^{n-1} + b,\ 0 \leq f_x \leq 1,\ \forall x \in B^n \right\}.$$

More precisely, it is proved that (8) and (Q) are equivalent in the sense that there exists $\tau_0 \geq 0$ such that for every $t > \tau_0$, the two problems have the same optimal value and the same set of optimal solutions.

3.3 DC Formulation

Let χ_K be the indicator function of K, say $\chi_K(f) = 0$ if $f \in K$, $+\infty$ otherwise. Since K is a convex set, χ_K is a convex function on $\mathbb{R}^{2^n}$.

A natural DC decomposition of the objective function of (Q) is

$$\Psi(f) + tp(f) := G(f) - H(f),$$

with $G(f) := \chi_{K(f)} + \Psi(f)$ and $H(f) := -tp(f)$. It is clear that G and H are convex functions. Thus Problem (Q) is a DC program of the form

$$(Q_{dc}) \quad \beta := \min\{G(f) - H(f) : \quad f \in \mathbb{R}^{2^n}\}.$$

Let ψ_w be the function defined by : $\psi_w(f) := \left| \sum_{x \in B^n} a_{wx} f_x - \frac{1}{2} \sum_{x \in B^n} a_{wx} \right|$. Then ψ_w is a convex function, and $\Psi(f) = \max_{w \in B^n} \psi_w(f)$. Therefore, Ψ is a polyhedral convex function. Likewise the function H is also polyhedral convex . So (Q_{dc}) is a polyhedral DC program where all DC decomposition are polyhedral. This property enhances DCA in the convergence theorem of DCA.

3.4 DCA to Solve (Q_{dc}) ([9])

Applying DCA to (Q_{dc}) amounts to computing, at each iteration k: $v^k \in \partial H(f^k)$ and $f^{k+1} \in \partial G^*(v^k)$. By the very definition of H, we can take v^k as follows:

$$v_x^k := -t \quad \text{if} \quad f_x^k \leq 0.5,\ t \quad \text{otherwise. .} \tag{9}$$

On the other hand, the computation of $f^{k+1} \in \partial G^*(v^k)$ is equivalent to the solution of the following linear program :

$$\min \left\{ \begin{array}{c} \xi - \langle v^k, f \rangle : \sum\limits_{x \in B^n} a_{wx} f_x - \frac{1}{2} \sum\limits_{x \in B^n} a_{wx} \leq \xi, \forall w \in B^n \\ - \sum\limits_{x \in B^n} a_{wx} f_x + \frac{1}{2} \sum\limits_{x \in B^n} a_{wx} \leq \xi, \forall w \in B^n, \\ 2^{n-1} - b \leq \sum\limits_{x \in B^n} f_x \leq 2^{n-1} + b, 0 \leq f_x \leq 1,\ \forall x \in B^n \end{array} \right\}. \tag{10}$$

The DCA applied to (Q_{dc}) can be described as follows:

DCA Algorithm
Let $f^0 \in \mathbb{R}^{2^n}$, and ϵ be a sufficiently small positive number.
Repeat
Set $v^k \in \partial H(f)$ via the formula (9);
Solving the linear program (10) to obtain f^{k+1};
Set $k := k + 1$.
Until $\| f^k - f^{k-1} \| < \epsilon$. Let us denote the feasible set of the linear program (10) and its vertex set by Ω and $V(\Omega)$, respectively. Let f^* be a solution computed by **DCA**. The convergence of **DCA** is summarized in the next theorem whose proof is essentially based on the convergence theorem of DCA applied to a polyhedral DC program.

Theorem 1. *(Convergence properties of Algorithm **DCA**)*

(i) ***DCA*** *generates a sequence* $\{f^k\}$ *contained in* $V(\Omega)$ *such that the sequence* $\{\Psi(f^k) + tp(f^k)\}$ *is decreasing.*
(ii) For a number t sufficiently large, if at iteration r we have $f^r \in \{0,1\}^{2^n}$*, then* $f^k \in \{0,1\}^{2^n}$ *for all* $k \geq r$*.*
(iii) The sequence $\{f^k\}$ *converges to* $\{f^*\} \in V(\Omega)$ *after a finite number of iterations. The point* f^* *is a critical point of Problem* (Q_{dc})*. Moreover if* $f^*_x \neq \frac{1}{2}$ *for all* $x \in B^n$*, then* f^* *is a local solution to* (Q_{dc})*.*

4 An Hybridation DCA-SA Approach

4.1 Simulated Annealing (SA) Procedure

In the SA method, each point s of the search space is analogous to a state of some physical system, and the function *E(s)* to be minimized is analogous to the system internal energy in that state. The goal is to direct the system from an arbitrary initial state to a state with possible minimum energy. At each step, the SA heuristic considers some neighbor state *s'* of the current state s, and probabilistically decides whether moving the system to the state *s'* or staying with the state s. The probabilities are chosen such that the system ultimately tends to move to states with lower energy. Typically the aforementioned process is repeated until a minimum temperature is reached. The technique has the following principal parameters:

- the temperature T
- the cooling rate $\alpha \in (0,1)$
- the number of moves N considered at each temperature cycle
- the maximum number $ICMax$ of temperature cycles considered before the search aborts.

In order to simultaneously exploit the efficiency of DCA and SA, we apply DCA at each iteration of SA when the current solution obtained by SA is accepted. The combined DCA-SA scheme can be described as follows.

4.2 A Combined DCA-SA Scheme

1. Let $T0$ be the start temperature.
2. Set $IC = 0$ (iteration count).
3. Randomly generate an initial current solution $\hat{f}_{curr}$.
4. **while** $(IC < ICMax)$
   {

   Repeat N Times
   {
   a- Generate $\hat{f}_{new} = neighbour\left(\hat{f}_{curr}\right)$.

{
b- Compute the difference in cost between $\hat{f}_{new}$ and $\hat{f}_{curr}$,
$\Delta_{cost} = cost\left(\hat{f}_{new}\right) - cost\left(\hat{f}_{curr}\right)$

c- If $(\Delta_{cost} < 0)$ then
{
c-1 Accept the move, $\hat{f}_{curr} = \hat{f}_{new}$.
c-2 Apply DCA procedure to improve the result.
}
d- Else
generate a value u from a uniform$(0, 1)$ random variable.
If $\left(\exp^{-\frac{\Delta_{cost}}{T}} > u\right)$
{
d-1 Accept the move,
d-2 Apply DCA procedure to improve the result.

}
otherwise reject it.
}

5. $T = T * \alpha$ $(The\ geometrical\ law\ of\ decrease)$**.**
6. $IC = IC + 1$.
}

7. The current value of $\hat{f}$ is taken as the final 'solution'.

We propose some variants to the combined algorithm which are based on the above algorithm.

- **SADCA1:** the DCA-SA scheme where the evaluation function is the objective function $\Psi(f)$ in (7).
- **SADCA2:** the DCA-SA scheme where the evaluation function is Clark's objective function

$$cost\left(\hat{f}\right) = \sum_{\omega \in F^n} \left| \left|\hat{F}(\omega)\right| - 2^{\frac{n}{2}} \right|^R, \quad R = 3. \tag{11}$$

- **Two phase DCA:** the DCA-SA scheme where **SADCA1** is applied in Phase 1 and DCA is applied in Phase 2 from the point obtained in Phase 1.

5 Experimental Results

In this section, we test the three variants of the combined DCA-SA scheme with the following schemes
SA: SA with the objective function $\Psi(f)$ in (7) as the evaluation function;

SAHC: Two stage approach of Clark and Jacob where SA is applied using the cost function in (4.1), then followed by a traditional hill-climb.

For the simulated annealing **SA**, the search is terminated either when 300 temperature cycles are reached or when 50 consecutive cycles do not produce an accepted move. At each temperature cycle, 400 moves are considered. The corresponding values for both Similarly, in the simulated annealing **Two phase DCA**, the search is stopped when either 45 temperature cycles are reached or 20 consecutive cycles do not produce an accepted move. 50 moves are considered at each temperature cycle. A temperature factor $\alpha = 0.9$ was used throughout. For DCA implementation, the parameters b, ϵ are taken to be equal to 0, and 10^{-6}, respectively. *Cplex 7.5* is used to solve the linear program. We have tested our approaches on a machine equipped with an *AMD Athlon 64 bits,dual core 3800+ (processor)* . The following tables show the best non-linearity achieved. We will compare our results with those presented in [4] by Clark and Jacob **SAHC**.

Careful observation reveals some interesting results.

– For $(n \geq 8)\, and\, (n < 12)$, although we considered only half of temperature cycles and moves as used in [4], **SADCA2** algorithm always gives the same

Table 1. Results for n=8

Technique	*CPU time*	*DCA iter*	*Non-linearity*
SA	1h 46min 10sec	-	114
SADCA1	53min 02sec	10	118
SADCA2	1h 12min 12sec	7	118
Two phase DCA	1h 20min	3	118
SAHC	-	-	116

Table 2. Results for n=9

Technique	*CPU time*	*DCA iter*	*Non-linearity*
SA	3h 27min 30sec	-	232
SADCA1	2h 26min 20sec	8	238
SADCA2	2h 53min 58sec	8	236
Two phase DCA	2h 54min 39sec	2	238
SAHC	-	-	236

Table 3. Results for n=10

Technique	*CPU time*	*DCA iter*	*Non-linearity*
SA	4h 13min 27sec	-	476
SADCA1	2h 58min 39sec	9	480
SADCA2	3h 33min 10sec	6	486
Two phase DCA	3h 40min 10sec	3	486
SAHC	-	-	484

Table 4. Results for n=11

Technique	*CPU time*	*DCA iter*	*Non-linearity*
SA	8h 34min 01sec	-	968
SADCA1	6h 49min 50sec	5	978
SADCA2	7h 34min 15sec	4	986
Two phase DCA	7h 58min 04sec	2	978
SAHC	-	-	984

Table 5. Results for n=12

Technique	*CPU time*	*DCA iter*	*Non-linearity*
SA	56h 46min 05sec	-	1984
SADCA1	22h 19min 01sec	6	1986
SADCA2	25h 15min 10sec	6	1984
Two phase DCA	24h 12min 06sec	3	1986
SAHC	-	-	1990

nonlinearity as in [4] and even superior in some cases ($n = 8$). Therefore, it is clear that using DCA in every temperature cycle improves the solution.
- In the **Two phase DCA** algorithm, only several iterations are needed to reach the best solution after DCA is restarted.
- All Boolean functions generated by our algorithm are balanced.

6 Conclusion

We have proposed an hybridation approach that combines DCA and SA. Numerical results show that the combined DCA-SA algorithm is much better than the classical SA scheme. This combined algorithm is very efficient to find a good initial point for DCA. Nevertheless, the two phase algorithm is the best among three versions of the hybridation approach. It means that, in any case, the role of DCA is crucial. In a future work we will improve the hybridation approach by an in-depth analysis of the two important issues : "when to restart DCA in the SA scheme" and "how many iterations of DCA needed to get a good solution in each cycle of SA". Computational experiments in the large-scale setting will also be performed in order to evaluate suitably the effectiveness and the efficiency of the algorithm.

References

1. Canteaut, A., Carlet, C., Charpin, P., Fontaine, C.: Propagation characteristic and correlation-immunity of hight nonlinenar Boolean function. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807. Springer, Heidelberg (2000)
2. Clark, J.A.: Optimisation Heuristics for Cryptology, PhD thesis, Faculty of Information Technology, Queensland University of Technology (1998)

3. Clark, J.A., Jeremy, J.L., Stepney, S.: The design of S-boxes by simulated annealing. In: CEC 2004: International Conference on Evolutionary Computation, Portland OR, USA, June 2004, pp. 1533–1537. IEEE, Los Alamitos (2004)
4. Clark, J.A., Jeremy, J.L.: Two-stage Optimisation in the Design of Boolean Functions. In: Proceedings of the 5th Australasian Conference on Information Security and Privacy. Lecture Notes In Computer Science, vol. 1841, pp. 242–254. Springer, Heildberg (2000)
5. Le Thi Hoai, A., Pham Dinh, T.: DC Programming: Theory, Algorithms and Applications. In: Proceedings of The First International Workshop on Global Constrained Optimization and Constraint Satisfaction (2002)
6. Le Thi Hoai, A., Pham Dinh, T.: The DC (difference of convex functions)Programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations Research (2005)
7. Le Thi Hoai, A., Pham Dinh, T., Huynh Van, N.: Exact Penalty Techniques in DC Programming Special session. In: Third International Conference on Computational Management Science, Amsterdam, May 17-19 (2006)
8. Carlet, C.: Boolean Functions for Cryptography and Error Correcting Code. Boolean Methods and Models. Cambridge University Press, Cambridge (2007)
9. Le Thi Hoai, A., Pham Dinh, T., Le Hoai, M., Bouvry, P.: A DC programming Approach for generating highly nonlinear balanced Boolean Function in Cryptography. In: Third International Conference on Computational Management Science, Amsterdam, May 17-19 (2006)
10. Le Thi Hoai, A., Pham Dinh, T., Le Hoai, M., Bouvry, P.: A combined DCA-GA for constructing highly nonlinear balanced Boolean functions in cryptography. In: Advances in Global Optimization, Myconos Greece, June 13-17 (2007)
11. Sarkar, S., Maitra, S.: Construction of nonlinear Boolean functions with important cryptographic properties. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 485–506. Springer, Heidelberg (2000)
12. Stallings, W.: Cryptography and Network Security, 3rd edn. Prentice Hall, Englewood Cliffs (2003)
13. Seberry, J., Zhang, X.M., Zheng, Y.: Nonlinearly balanced Boolean functions and their propagation characteristics. In: Advances in Cryptographie - EUROCRYPT 2000 In Advances in Cryptology - CRYPT0 1993. Springer, Heidelberg (1994)

Non-standard Attacks against Cryptographic Protocols, with an Example over a Simplified Mutual Authentication Protocol

Julio C. Hernandez-Castro, Juan M.E. Tapiador, and Arturo Ribagorda

Department of Computer Science, Carlos III University of Madrid
{jcesar,jestevez,arturo}@inf.uc3m.es

Abstract. In this work, we present a simple model for the automated cryptanalysis of cryptographic protocols based on meta-heuristic search. We illustrate our approach with a straightforward application in the form of an attack against a slightly simplified version of an ultra-lightweight authentication protocol for RFID environments called SASI. We show how an attack based on Simulated Annealing can efficiently recover the tag's secret ID, which is the value the protocol is designed to conceal.

1 Introduction

In recent years there has been a proliferation of cryptographic protocols aimed at providing security services in very constrained environments. The most common examples are mutual authentication schemes for Radio Frequency IDentification (RFID) systems, where the shortage of computational resources in tags makes impossible to apply classical constructions based on cryptographic primitives such as block/stream ciphers or hash functions.

Typical proposals are often forced to consist of a number of steps in which very simple operations are performed over public and private values. In this work, we present a general model for the automated cryptanalysis of such schemes based on meta-heuristic search. We will illustrate our idea with an application against a simplified version of one of the most prominent protocols proposed so far: an ultra-lightweight authentication protocol for RFID environments called SASI. We will be able to show how an attack based on a Simulated Annealing technique can efficiently recover the tag's secret ID, which is the value the protocol is designed to conceal.

The idea of attacking cryptographic protocols by means of heuristic procedures is not new: Clark et al. [4] presented a seminal work in this area where they were able to break the PPP [8] identification protocol. It is, however, true that since that work no further progress has been made in the field. Additionally, the technique employed here is quite different from that used by Clark et al. The related area of evolving or automatically designing cryptographic protocols by means of different heuristic techniques has seen, on the other hand, considerable success [5].

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 589–596, 2008.

The rest of the paper is organized as follows. In the next section, we present our general model proposal for non-standard attack of cryptographic protocols. After this, in Section 3 we describe a novel authentication protocol for RFID environments called SASI, together with its simplified variant CR-SASI, which succumbs to the attack introduced and analyzed in Section 4. Finally, in Section 5 we extract some conclusions.

2 General Attack Model

The main idea behind our approach is to transform the cryptanalysis of a security protocol into a search problem, where search heuristics can be applied. In general, during this search what we will try to find are the secret state values (keys, nonces, etc.) of some subset of the involved parties. This, of course, could be done in various ways, but the most natural approach is to measure the fitness of the tentative secret values by the proximity of the messages produced by these tentative solutions to the real public messages generated and exchanged during the actual protocol execution.

Most cryptographic protocols should exchange one or more messages to accomplish their intended objective (authentication, key exchange, key agreement, etc.), and in the vast majority of cases these messages are sent via an insecure or public channel that can be easily snooped.

In our attack model, the cryptanalyst will try to infer the secret values that the two parties are trying to hide by exploiting the knowledge of the exchanged messages. In an robust, secure and well-designed cryptographic protocol, even states that are very close to the real state should not produce messages that are very close (for any useful distance definition) of the real public messages.

This should be done, typically, by means of a carefully design and message construction based on the use of some highly nonlinear cryptographic primitives such as block ciphers or hash functions. Weaknesses in a protocol design could lead, on the other hand, to the lack of this desirable property, a fact that can be exploited to mount a non-standard cryptanalytic attack based in some kind of heuristic search guided by a fitness measuring distances between exchanged and computed messages.

This shall exactly be the approach followed in the rest of the paper.

3 Description of the SASI Protocol

In 2007, Chien presented an interesting ultra-lightweight mutual authentication protocol providing Strong Authentication and Strong Integrity (SASI), intended for very low-cost RFID tags [1]. This was a much needed answer to the increasing need for schemes providing such properties in extremely constrained environments like RFID systems. As all the previous attempts to design ultra-lightweight protocols have failed (essentially all proposals have been broken), this new scheme was specially interesting.

The SASI protocol is briefly described in the following, where R represents the reader, T represents the tag, IDS stands for an index pseudonym, ID is tag's private ID, K_i represent tag's secret keys and n_1 and n_2 are nonces. Each message takes the form $A \rightarrow B : m$, meaning that sender A sends to receiver B message m.

1. $R \rightarrow T : hello$
2. $T \rightarrow R : IDS$
3. The reader uses IDS to find in the back-end database the tag's secret values ID, K_1, and K_2. Then R generates nonces n_1 and n_2 to construct messages A, B and C as follows:
   $A = IDS \oplus K_1 \oplus n_1$
   $B = (IDS \vee K_2) + n_2$
   $C = (K_1 \oplus \bar{K_2}) + (K_2 \oplus \bar{K_1})$, where
   $\bar{K_1} = Rot(K_1 \oplus n_2, K_1)$
   $\bar{K_2} = Rot(K_2 \oplus n_1, K_2)$
   where $\oplus$ stands for the usual addition modulo 2, $+$ represents addition modulo 2^{96}, $Rot(A, B) = A << wht(B)$ with $wht(B)$ the Hamming weight of B, and $\vee$ is the usual bitwise or operation. Finally, the reader sends to the tag the concatenation of A, B and C:

   $R \rightarrow T : A \| B \| C$
4. From A and B,respectively, the tag can obtain values n_1 and n_2. Then it locally computes C and checks if the result of its local computation is equal to the sent value. If this is the case, it updates the values of IDS, K_1 and K_2 in the following manner:

   $IDS^{next} = (IDS + ID) \oplus (n_2 \oplus \bar{K_1})$
   $K_1^{next} = \bar{K_1}$
   $K_2^{next} = \bar{K_2}$
5. $T \rightarrow R : D$, where $D = (\bar{K_2} + ID) \oplus ((K_1 \oplus K_2) \vee \bar{K_1})$
6. Finally, R verifies D and, if it is equal to the result of its local computation, it updates IDS, K_1 and K_2.

SASI has received no serious attacks yet, except for a couple of minor weaknesses that could be employed to mount two desynchronization scenarios [2].

3.1 CR-SASI: The Simplified SASI Protocol

CR-SASI is the simplified version of the SASI protocol we will cryptanalyze in the paper. It is essentially identical to the published version, but for two minor differences:

1. CR-SASI is a scaled-down version of SASI, which operates over $\mathbb{Z}_2^{32}$ while SASI does it over elements in $\mathbb{Z}_2^{96}$
2. CR-SASI uses constant distance rotations, instead of the Hamming-dependant rotations proposed by Chien.

Any amount of non-zero rotation produces a similarly robust protocol, but for reasons explained below we have fixed this rotation amount to $\frac{32}{2} = 16$. We have experimentally found, anyway, that any other value produces a protocol that is breakable in essentially the same way. An important observation is that the amount of the rotation operation, as originally defined, is far from being uniform. In fact, if we assume this second argument B to be random, then the probability that the rotation amount takes value k is given by the formula:

$$Prob(wht(B) = k) = \frac{\binom{32}{k}}{2^{32}} \tag{1}$$

which attains a maximum for $k = \frac{32}{2} = 16$ with an associated probability of 0.139949, or around 14% of the times. This additionally justifies our chosen value (as it will be the most common) for the amount of left rotation in CR-SASI.

All in all, these two modifications should not greatly modify the security characteristics of the underlying protocol, so the study of this variant is relevant for understanding the security of the whole SASI protocol. It is important to notice that constant distance rotations, such as that used in CR-SASI, are usually part of cryptographic primitives and protocols, so they could have been part of the original protocol proposal as they were components of modern cryptographic primitives such as TEA[6], XTEA[9] and Salsa20[7], to name a few.

4 Cryptanalysis of the SASI Protocol

In the light of the equations that define the protocol (see Section 3) we can initially see that the internal secret state we will look for is formed of the values:

$$State = \{K_1, K_2, n_1, n_2, ID\}$$

Assuming that message A is known (as it is the case), it can be seen than K_1 and n_1 are related ($n_1 = A \oplus IDS \oplus K_1$), so we can reduce the state size to $\{K_1, K_2, n_2, ID\}$.

Analogously, from the knowledge of message B we can conclude that K_2 and n_2 are also related, following the equation $n_2 = B - (IDS \vee K_2)$. We can therefore still reduce the state size to $State = \{K_1, K_2, ID\}$.

A further reduction is still possible, since ID also depends on $\{K_1, K_2\}$, although in a more complex way, once $\{IDS^{next}, IDS\}$ or D are known because:

$$ID = IDS^{next} \oplus (n_2 \oplus \bar{K_1}) - IDS \tag{2}$$

$$ID = D \oplus ((K_1 \oplus K_2) \vee \bar{K_1}) - \bar{K_2} \tag{3}$$

So we finally are left with a minimal state of the form $State = \{K_1, K_2\}$, where no further reduction is possible. This implies that our set of possible solutions are of the above described form, and have a size of 2^{64}.

Note that among the public messages $\{IDS, A, B, C, D, IDS^{next}\}$ that can be observed after one authentication session, we have used all except C and D to reduce the state space. These two last messages will be the base of our fitness function.

Starting from a candidate state $\{K_1', K_2'\}$, and using public values $\{IDS, A, B, IDS^{next}\}$, we will compute the corresponding values for messages C' and D', and measure their distance to the known actual values of C and D. We will try to minimize this distance in order to find values as close as possible to the real $\{K_1, K_2\}$ values. Different definitions of distance have been tried (euclidean, edit, weighted, etc.) and could be useful, but we have got the best results with the usual Hamming distance. In the general case of having N public messages $M_1, \ldots, M_N$, the resulting fitness function is given by:

$$f_S = -\sum_{i=0}^{N} d_H(M_i, M_i') \tag{4}$$

where M_i stands for the real (snooped) message and M_i' is its approximation as computed from the values of state S.

For our particular problem, equation (4) will have the form:

$$f_S = -\big(d_H(C, C') + d_H(D, D')\big) \tag{5}$$

For this optimization process, we will use Simulated Annealing as heuristic. After extensive experimentation, the set of parameters which consistently lead to good results are those shown in Table 1.

4.1 Experimental Results

We have implemented various versions of the same cryptanalytic method, which start the cryptanalytic process after eavesdropping two, three or four consecutive rounds of the protocol, respectively. As expected, the knowledge of more authentication rounds leads to better attacks.

We have performed simulations for measuring the effectiveness of this approach. In all the cases, we initialized all secret and public values of the protocol to the first hexadecimal values of π, as taken from `http://www.super-computing.org/`. In particular, the state values we are looking for are fixed to $State = \{0x243F6A88, 0x85A308D3\}$.

Results for five different runs, after capturing data from only two consecutive authentication sessions are shown in Table 2.

It can be clearly seen that, although no solution is perfect, all of them are quite close to the real secret state values. In fact, it is very easy to compute

Table 1. Simulated Annealing parameters for the cryptanalysis of CR-SASI

Initial Temperature	10
Cooling Rate	0.99
Max. Failed Cycles	100
Moves	1000
IC Max.	500

the secret from these accurate approximations. Just a bitwise majority function (weighted by the fitness as provided in Table 2) will lead to an almost correct solution. Then, a simple trial and error process could be started to find the correct values. In case of any doubt, more runs should improve the accuracy of the attack.

We have experimentally found that about four or five runs are generally enough to get results very close to the secret values, as in the case of the above example. At worst, only 16^2 additional trials are needed for recovering the correct keys.

Each of these runs takes approximately 800 seconds in a very modest laptop. Furthermore, it is important to stress that they are completely parallelizable. After obtaining the correct key values, the secret static ID will be easily recovered using either equation (2) or (3). This will allow a fraudulent tag (with, say, altered prices or false stock information) to impersonate the legitimate tag, possibly corrupting the back-end database with false data, after eavesdropping only two consecutive authentication sessions.

The efficiency of this attack can be slightly improved just by observing more sessions. We have performed tests after three and four consecutive sessions following exactly the same approach described above, and the results were consistently better.

However, with more authentication sessions there are approaches that do not work after eavesdropping only two sessions that now become entirely possible. In this case, the best attacking strategy becomes to try to infer the secret ID from the best approximation found for $\{K_1, K_2\}$ with the help of equations (6)–derived from (2))– and (7):

$$ID = IDS^{next} \oplus (n_2 \oplus \bar{K}_1) - IDS \tag{6}$$

$$= IDS^{next} \oplus ((B - (IDS \vee K_2')) \oplus \bar{K}_1') - IDS \tag{7}$$

After 10 runs following this scheme, we obtained the exact value of the secret ID three times (see Table 3), and very good approximations to it (with a Hamming distance to the real ID of 8 or less) in another six occasions. The best approximation was very easy to identify because it has the best fitness within the 10 runs.

Table 2. Attack results for five runs, after capturing two authenication sessions

K1=0x24FF6B8E	K2=0x84E308D5	Fitness=-7.000000
K1=0x74300A88	K2=0x35ACA8C3	Fitness=-7.000000
K1=0x347FCA88	K2=0x35E368D3	Fitness=-6.000000
K1=0x343FCAC8	K2=0x35E36893	Fitness=-4.000000
K1=0x243F2A88	K2=0x85A348D3	Fitness=-1.000000
K1=0x243F?A88	K2=0x85A3?8D3	Majority weighted function
K1=0x243F6A88	K2=0x85A308D3	Real Values

Table 3. Attack results for 10 runs, after capturing 4 authentication sessions

Fitness	d_H to ID
-21	0
-27	0
-26	0
-23	5
-33	9
-31	7
-31	3
-24	5
-33	8
-27	3

5 Concluding Remarks

In this paper, we have presented a new and efficient attack against a simplified version of a novel and interesting ultra-lightweight authentication protocol.

This attack is performed by means of a non-standard technique (SA-based) that we have shown as a particular instance of a more general attack methodology against cryptographic protocols.

We believe that more and more of these non-standard attacks will be sucessfully employed against the new lightweight protocols designed for very constrained environments such as RFID systems and some kinds of sensor networs, because in most of the cases they can't allow the use of standard cryptographic primitives.

Attacking the full SASI protocol with similar but improved techniques is a future and interesting research direction.

References

1. Chien, H.-Y.: SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. IEEE Transactions on Dependable and Secure Computing 4(4), 337–340 (2007)
2. Sun, H.-M., Ting, W.-C., Wang, K.-H.: On the Security of Chien's Ultralightweight RFID Authentication Protocol. Cryptology ePrint Archive, http://eprint.iacr.org/2008/083
3. Klimov, A., Shamir, A.: New Applications of T-functions in Block Ciphers and Hash Functions. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557. Springer, Heidelberg (2005)
4. Clark, J.A., Jacob, J.L.: Fault Injection and a Timing Channel on an Analysis Technique. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 181–196. Springer, Heidelberg (2002)
5. Clark, J.A., Jacob, J.L.: Protocols are Programs Too: the Meta-heuristic Search for Security Protocols. Metheuristics for Software Engineering. Information Software Technology 43(14), 891–904 (2001)

6. Wheeler, D.J., Needham, R.M.: TEA, a tiny encryption algorithm. In: Fast Software Encryption: Second International Workshop, Leuven, Belgium. LNCS, vol. 1008, pp. 363–366. Springer, Heidelberg (1994)
7. Bernstein, D.J.: The Salsa20 stream cipher, slides of talk at ECRYPT STVL Workshop on Symmetric Key Encryption (2005), http://cr.yp.to/talks.html#2005.05.26
8. Pointcheval, D.: A New Identification Scheme Based on the Perceptron Problems. In: Advances in Cryptology Eurocrypt 1995. LNCS, vol. 2199. Springer, Heidelberg (1995)
9. Needham, R.M., Wheeler, D.J.: Tea extensions. Technical report, Computer Laboratory. University of Cambridge, Cambridge (October 1997)

Provable Security against Impossible Differential Cryptanalysis Application to CS-Cipher

Thomas Roche[1,2,7], Roland Gillard[5,2,3], and Jean-Louis Roch[1,3,4,6]

[1] Laboratoire d'Informatique de Grenoble, 51 av. Jean Kuntzmann, 38330 Montbonnot-Saint-Martin, France
[2] Université Joseph Fourier
[3] Grenoble Université
[4] INRIA Rhône-Alpes
[5] Institut Fourier, 100 rue des Maths, BP74 38402 St Martin d'Hères, France
[6] Institut National Polytechnique de Grenoble (INPG)
[7] CS, Communication&Systems, 22 avenue Galilée, 92350 Le Plessis Robinson, France
Thomas.Roche@imag.fr, Roland.Gillard@ujf-grenoble.fr, Jean-Louis.Roch@imag.fr

Abstract. In this document we present a new way to bound the probability of occurrence of an n-round differential in the context of differential cryptanalysis. Hence this new model allows us to claim proof of resistance against impossible differential cryptanalysis, as defined by Biham and al. in 1999. This work will be described through the example of CS-Cipher, to which, assuming some non-trivial hypothesis, provable security against impossible differential cryptanalysis is obtained.

Keywords: Impossible Differential cryptanalysis, Provable security, Symmetric ciphers

1 Introduction

The resistance against differential cryptanalysis has been studied since the attack invention by Biham and Shamir (1990 [1]). Formal proofs based on the Markov cipher approximation (Lai and Massey [2]) and related to the minimal number of active S-Boxes in a differential characteristic are now well known. On the other hand, it is hardly possible to evaluate a symmetric cipher w.r.t. impossible differential cryptanalysis. Inspired by the work of Sugita and al. in [3], we are going to introduce a new way to approach the probability of occurrence of an n-round differential. Although this approach does not give better upper bound than has already been done, it allows us to display a lower bound and then claim resistance against impossible differential for an example cipher. The study focuses on CS-Cipher (symmetric cipher introduced by Stern and Vaudenay in [4]); its resistance against differential and truncated differential cryptanalysis has been studied in [5]. As in [5] we will use the properties of CS-Cipher multipermutations in order to decrease the complexity of computing our bounds.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 597–606, 2008.

Let us note that our proof holds on the hypothesis that the symmetric cipher is a Markov cipher and a Support Markov cipher (notion about to be introduced in this document) with uniformly distributed round keys.

Notations and Material

An iterated or block cipher performs a sequence of rounds to encrypt a plaintext of fixed size (block size). In all the sequel, the following notations and material are used with respect to an iterated or block cipher.

n, m: Denotes respectively the block size in bits and in bytes (i.e. $n = 8 \times m$).

$\oplus$: Denotes a group operation over the Galois field $GF(2)^8$ (in all the sequel this operation will be the bitwise addition modulo 2).

$\Delta_{\mathbf{x}}(\mathbf{x}')$: Denotes the *difference* between x and x' by the $\oplus$ operation. $x \oplus x' = \triangle_x(x')$. Noted $\triangle x$ when not ambiguous.

***i*-round Output ($\mathbf{O_i(x)}$):** Let x be a plaintext input of the cipher; $O_i(x)$ denotes the output after the i^{th} round.

***i*-round Differentials:** For an iterated cipher, a pair (α, β) is a possible i-round differential if and only if there is a pair of plaintext input (x, x') such that $x \oplus x' = \alpha$ and $O_i(x) \oplus O_i(x') = \beta$. Later on, a 1-round differential is simply called a *differential*.

***i*-round Characteristics:** For an iterated cipher, a set $\Omega = \{\omega_0, \omega_1, \ldots, \omega_i\}$ is a possible i-round characteristic if and only if there is a pair of plaintext input (x, x') such that $x \oplus x' = \omega_0$ and $\forall j \in \{1 \ldots i\}, O_j(x) \oplus O_j(x') = \omega_j$. Hence, an i-round characteristic is a sequence of i j-round differentials with $j \in \{1, \ldots, i\}$.

Probability of a differential (DP^f): Given a boolean function $f : GF(2)^p \longrightarrow GF(2)^q$, for any $\alpha \in GF(2)^p$ and any $\beta \in GF(2)^q$ we note :

$$DP^f(\alpha, \beta) = \Pr_x\{x | f(x) \oplus f(x \oplus \alpha) = \beta\}$$

S-Boxes: Substitution boxes are fairly common in block ciphers, they are functions that give the necessary non-linearity of encryption functions. The non-linearity with respect to differential cryptanalysis is evaluated by computing the DP^{S-Box}.

Active S-Boxes for a given characteristic (or differential) are the encryption function's S-Boxes that present a non null difference for input.

Multipermutations: The notion of multipermutation was introduced by Schnorr and Vaudenay in [6]. For our needs in this paper we will just define the general idea of a $(2, 2)$-multipermutation over $GF(2)^8$, of which complete description can be found in Vaudenay's PhD thesis ([7]). A $(2, 2)$-multipermutation over $GF(2)^8$ can be seen as a permutation over $GF(2)^{16}$ such that fixing the first half of the input (respectively the second part) makes both half of the output permutations of the second half of the input (respectively the first part).

Markov Chain: A sequence of discrete random variables $(X_r, \ldots, X_0)$ forms a Markov chain if and only if : $\forall i \in \{0 \ldots r-1\}$,

$$\Pr(X_{i+1} = x_{i+1} | X_i = x_i, \ldots, X_0 = x_0) = \Pr(X_{i+1} = x_{i+1} | X_i = x_i)$$

Markov Ciphers: Denotes a subclass of iterated ciphers, first introduced by Lai, Massey and Murphy in [2] to give a formal environment to iterated ciphers and then lead to provable security against differential cryptanalysis. An r-round iterated cipher is a Markov cipher when the sequence $(\triangle x = \triangle y_0, \triangle y_1, \ldots, \triangle y_r)$ of round output *differences* forms a Markov chain. That is to say

$$\Pr(\triangle y_r = \omega_r | \triangle y_0 = \omega_0, \triangle y_1 = \omega_1, \ldots, \triangle y_{r-1} = \omega_{r-1}) = \Pr(\triangle y_r = \omega_r | \triangle y_{r-1} = \omega_{r-1})$$

CS-Cipher

CS-Cipher was introduced by Jacques Stern and Serge Vaudenay in [4]. In this section we will just introduce its main characteristics. For more information, the reader can refer to the original description.

CS-Cipher is an iterated block cipher of 64 bits block size, and 128 bits key size. It consists of 8 iterations of a round function E followed by a bit-width XOR operation ($\oplus$) with the last 64-bits round key.

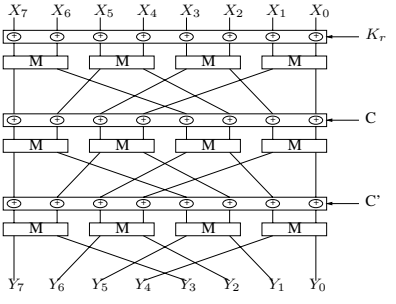


Fig. 1. CS-Cipher round Block diagram. Function E.

Round Description The Figure 1 presents one round of CS-Cipher. The XORed values K_r, C and C' are respectively the 64-bits round key, a first and a second constant.

By definition, $M(x, y) = (\mu(P(x), P(y))$ (see Figure 2) , the functions μ and P being respectively a $(2, 2)-$*multipermutation* over $GF(2)^8$ and a *non-linear* permutation over $GF(2)^8$. They are defined as follows:

- $\mu(a, b) = (\varphi(a) \oplus b, R_l(a) \oplus b)$. Where R_l is a $1-$bit shift circular rotation to the left and φ is defined by $\varphi(x) = (R_l(x) \wedge 0x55) \oplus x$ where $\wedge$ represents the bitwise AND. Hence the input/output pattern around a μ box will follow one out of those six patterns (Stars meaning any non-zero values):

$$\mu(0, 0) = (0, 0), \mu(*, 0) = (*, *), \mu(0, *) = (*, *)$$
$$\mu(*, *) = (*, *) \text{ or } (*, 0) \text{ or } (0, *)$$

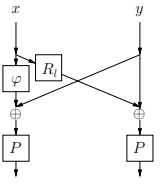


Fig. 2. CS-Cipher M box

– P, defined by a 256-elements table, is CS-Cipher S-Box. Let us give upper and lower bounds of P's differential probability :

$$DP_{max} = \max_{a \neq 0, b} DP^P(a, b) \leq 2^{-4}$$
$$DP_{min} = \min_{a,b} DP^P(a, b) \geq 2^{-7}$$

These values are easy to compute, one has just to compute all the possible values of $DP^P(a, b)$ for any value (a, b) (there are 2^{16} pairs).

Differential and Linear Cryptanalysis. In [5], Serge Vaudenay gives sufficient arguments to heuristically prove the security of CS-Cipher against differential and truncated differential (when considering characteristics and not simple differential). The formal treatment of differential cryptanalysis based on Markov cipher is not detailed in the present document, please refer to [2] for a more complete description.

Considering the probabilistic event :

$$E_{\omega_i, \omega_0} : \{O_i(x) \oplus O_i(x') = \omega_i \mid x \oplus x' = \omega_0\},$$

where (x, x') are two plaintexts.

Randomly chosen plaintexts pair of difference ω_0 will create an output difference ω_i after i rounds with probability $\Pr_{x,x'}(E_{\omega_i,\omega_0})$. Differential cryptanalysis works when one can find (ω_0, ω_i) for which the value $\Pr_{x,x'}(E_{\omega_i,\omega_0})$ is "high".

Vaudenay proves that CS-Cipher is immune against any cryptanalysis using statistics over differential characteristics which have more than 2 rounds. The author can then claim immunity against all kind of differential attacks when CS-Cipher has more than 4 rounds. Finally the study of resistance against truncated differential, which corresponds to group sets of characteristics in order to improve differential cryptanalysis, is evaluated to be strong enough after 5.33 rounds.

Impossible Differential Cryptanalysis. This type of attack was introduced by Biham, Biryukov and Shamir in 1999 in [8]. From [8], in an Impossible differential attack, *"a differential predicts that particular differences should not occur (i.e., that their probability is exactly zero), and thus the correct key can never decrypt a pair of ciphertexts to that difference. Therefore, if a pair is decrypted*

to this difference under some trial key, then certainly this trial key is not the correct key. This is a sieving attack which finds the correct keys by eliminating all the other keys which lead to contradictions.“

CSC*. For purpose of clarity, we are going to consider a slightly different cipher than CS-Cipher, CSC*. This variant was introduced by Vaudenay in [5] in order to simplify the proof of resistance. In CSC* the key schedule is replaced by a true random generator of 25 64-bits round keys. Hence the CS-Cipher round keys are replaced by 9 CSC* round keys and each XOR to constants C or C' is replaced by a XOR to one of the CSC* round keys. The new cipher CSC* can then be seen as a 24 rounds block cipher with a simple round function (see Figure 3). The results found in [5] for CSC* are believed to hold for CS-Cipher, and in this document we make the same assumption.

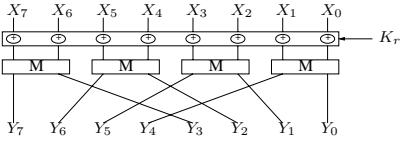


Fig. 3. CSC* round Block diagram

Notations. In all the sequel, we will use [5]'s notations to describe CSC* components, thus the i^{th} round of CSC* can be written as follow:

$$\rho_i = L_\pi \circ P^8 \circ \mu^4 \circ s_{i-1}$$

where, for any 64-bits element $x = (x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$,

- $s_{i-1}(x) = x \oplus K_i$ (K_i is the i^{th} round key)
- $\mu^4(x) = (\mu(x_7, x_6), \mu(x_5, x_4), \mu(x_3, x_2), \mu(x_1, x_0))$
- $P^8(x) = (P(x_7), P(x_6), P(x_5), P(x_4), P(x_3), P(x_2), P(x_1), P(x_0))$
- $L_\pi(x) = (x_7, x_5, x_3, x_1, x_6, x_4, x_2, x_0)$

Hence, the block encryption CSC* can be written as:

$$Enc = s_{24} \circ \rho_{24} \circ \ldots \circ \rho_1.$$

2 Output Differential

In this section we are going to introduce the notion of *Support Markov Cipher* and show that under the hypothesis of Markov Cipher, Support Markov Cipher and uniformly distributed round keys it is possible to display a lower bound of r-round differential probability. Then, as an example we will apply this proof to CS-Cipher and show that it is indeed resistant against impossible differential.

2.1 Formal Treatment for CSC*

Note: All probabilities are average probabilities over the key distribution (which is assumed to be uniform).

Definition 1. *The support function χ (referred as the characteristic function in [3])*

$$\chi : (GF(2)^k)^m \to (GF(2))^m, (x_0, \ldots, x_m) \longrightarrow (y_0, \ldots, y_m)$$

such that

$$y_i = \begin{cases} 0 \text{ if the } k\text{\_uplet } x_i = 0, \\ 1 \text{ otherwise.} \end{cases}$$

Remark : for CS-Cipher and CSC, $k = m = 8$.*

Lemma 1. *Let us consider a plaintext pair (x, x') such that $x \oplus x' = \triangle y_0$ and the output differences $(\triangle y_r, \ldots, \triangle y_0)$ generated by an encryption of x and x' by CSC*. We have for any i in $\{0, \ldots, r-1\}$:*

$$\chi(\triangle y_{i+1}) = \chi(L_\pi \circ \mu^4(\triangle y_i)).$$

Proof. The proof, easy to obtain, is provided in an online version of this paper.

Definition 2. *An r-round iterated cipher is a* Support Markov Cipher *when the sequence $(\chi(\triangle x = \triangle y_0), \chi(\triangle y_1), \ldots, \chi(\triangle y_r))$ of round output* differences *support forms a Markov chain.*

Hereafter, in order to simplify the formulas, the sequence round output *differences* as random variables will be referred as the sequence $(X_r, \ldots, X_0)$ instead of $(\triangle y_r, \ldots, \triangle y_0)$.

Lemma 2. *Let us consider a Markov cipher E and its associated Markov chain $(X_r, X_{r-1}, \ldots, X_1, X_0)$, we have trivially:*

$$Pr(X_1 = x_1 \mid X_0 = x_0) \leq DP_{max}^{h(x'_1)} Pr(\chi(X_1) = x'_1 \mid X_0 = x_0),$$

where $h : (GF(2))^m \to \{0, \ldots, m\}$ gives the Hamming weight.

Lemma 3. *Let us consider a Markov cipher E and its associated Markov chain $(X_r, X_{r-1}, \ldots, X_1, X_0)$, we have trivially:*
if $Pr(X_1 = x_1 \mid X_0 = x_0) \neq 0$ then

$$Pr(X_1 = x_1 \mid X_0 = x_0) \geq DP_{min}^{h(x'_1)} Pr(\chi(X_1) = x'_1 \mid X_0 = x_0),$$

where $h : (GF(2))^m \to \{0, \ldots, m\}$ gives the Hamming weight.

Theorem 1. *Let us consider CSC* as a Markov cipher and a Support Markov cipher E and its associated Markov chains $(X_r, X_{r-1}, \ldots, X_0)$.*

$$\begin{aligned} Pr(X_r = x_r \mid X_0 = x_0) \leq\ & [DP_{max} \times (2^8 - 1)]^{h(x'_1)} \\ & \times Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1) \\ & \times Pr(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1), \end{aligned}$$

where $h : (GF(2))^m \to \{0, \ldots, m\}$ gives the Hamming weight.

Proof. From the probability total formula

$$Pr(X_r = x_r \mid X_0 = x_0) = \sum_{x_1} Pr(X_r = x_r \mid X_1 = x_1, X_0 = x_0) \times Pr(X_1 = x_1 \mid X_0 = x_0)$$

From Lemma 2 and the fact that $(X_r, X_{r-1}, \ldots, X_0)$ is a Markov chain, we have

$$Pr(X_r = x_r \mid X_0 = x_0) \leq DP_{max}^{h(x'_1)} \times \sum_{x_1} [Pr(X_r = x_r \mid X_1 = x_1) \times Pr(\chi(X_1) = \chi(x_1) \mid X_0 = x_0)]$$

From Lemma 1 we have $\chi(X_1) = \chi(L_\pi \circ \mu^4(X_0))$ and then
$Pr(\chi(X_1) = \chi(x_1) \mid X_0 = x_0) = \begin{cases} 1 \text{ if } \chi(x_1) = \chi(L_\pi \circ \mu^4(x_0)) \\ 0 \text{ otherwise} \end{cases}$
Let us set $x'_1 = \chi(L_\pi \circ \mu^4(x_0))$, we have

$$Pr(X_r = x_r \mid X_0 = x_0) \leq DP_{max}^{h(x'_1)} \times \sum_{\substack{x_1 s.t. \\ \chi(x_1) = x'_1}} \frac{1}{Pr(X_1 = x_1)} \times Pr(X_r = x_r \ \& \ X_1 = x_1)$$

And since $Pr(X_1 = x_1)$ is a constant over all values of x_1, we have

$$Pr(X_r = x_r \mid X_0 = x_0) \leq DP_{max}^{h(x'_1)} \times \frac{Pr(\chi(X_1) = x'_1)}{Pr(X_1 = 0)} \times Pr(X_r = x_r \mid \chi(X_1) = x'_1)$$

Let us now introduce $\chi(X_r)$ in the equation

$$Pr(X_r = x_r \mid X_0 = x_0) \leq [DP_{max} \times (2^8 - 1)]^{h(x'_1)} \times \sum_{x'_r} Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1) \times Pr(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1)$$

And since $Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1) = \begin{cases} 1 \text{ if } \chi(x_r) = x'_r \\ 0 \text{ otherwise} \end{cases}$
Let us set $x'_r = \chi(x_r)$

$$Pr(X_r = x_r \mid X_0 = x_0) \leq [DP_{max} \times (2^8 - 1)]^{h(x'_1)} \times Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1) \times Pr(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1)$$

Theorem 2. *Let us consider CSC* as a Markov cipher and a Support Markov cipher E and its associated Markov chains $(X_r, X_{r-1}, \ldots, X_0)$.*

$$Pr(X_r = x_r \mid X_0 = x_0) \geq [DP_{min} \times 2^{-4} \times (2^8 - 1)]^{h(x'_1)} \times Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1) \times Pr(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1)$$

where $h : (GF(2))^m \rightarrow \{0, \ldots, m\}$ gives the Hamming weight.

Proof. As in the proof of Theorem 1, let us first introduce X_1 in the equation

$$Pr(X_r = x_r \mid X_0 = x_0) = \sum_{x_1} Pr(X_r = x_r \mid X_1 = x_1, X_0 = x_0) \times Pr(X_1 = x_1 \mid X_0 = x_0)$$

From Lemma 3 and the fact that $(X_r, X_{r-1}, \ldots, X_0)$ is a Markov chain, we have

$$Pr(X_r = x_r \mid X_0 = x_0) \geq DP_{min}^{h(x'_1)} \times \sum_{\substack{x_1, \text{ s.t.} \\ DP(\mu^4(x_0), L_\pi^{-1}(x_1)) \neq 0}} Pr(X_r = x_r \mid X_1 = x_1)$$

Let us set $\begin{cases} Poss_{x_0} = \{x, \text{ s.t. } DP(\mu^4(x_0), L_\pi^{-1}(x)) \neq 0\} \\ Supp_{x_0} = \{x, \text{ s.t. } \chi(x) = \chi(L_\pi \circ \mu^4(x_0))\} \end{cases}$

We are now going to estimate the value of

$$\sum_{x_1 \in Poss_{x_0}} Pr(X_r = x_r \mid X_1 = x_1) \text{ w.r.t. } \sum_{x_1 \in Supp_{x_0}} Pr(X_r = x_r \mid X_1 = x_1).$$

One can easily note that $Poss_{x_0} \subset Supp_{x_0}$ and from CSC* characteristics,

$$Card(\{Poss_{x_0}\}) \geq (2^{-4})^{h(\chi(\mu^4(x_0)))} Card(\{Supp_{x_0}\})$$

From the markovian property of the chain $(X_r, X_{r-1}, \ldots, X_0)$, the value of $Pr(X_r = x_r \mid X_1 = x_1)$ is independent to the fact that $x_1 \in Poss_{x_0}$ or $x_1 \in Supp_{x_0}$. Finally, we have

$$Pr(X_r = x_r \mid X_0 = x_0) \geq [DP_{min} \times (2^8 - 1)]^{h(x'_1)} \times (2^{-4})^{h(x'_1)} \times Pr(X_r = x_r \mid \chi(X_1) = x'_1)$$

The proof ends exactly like in Theorem 1.

2.2 Results for CSC*/CS-Cipher

Let us assume an uniform distribution of the round keys and that CSC* and CS-Cipher can be considered as Markov Ciphers and Support Markov Ciphers.

Theorem 1 and Theorem 2 give an upper and lower bound for the probability of occurrence of a r-round differential.

In order to evaluate these bounds, we have to approach the two values $Pr(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1)$ and $Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1)$.

1. $Pr(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1)$. By definition, the set $(\chi(X_r), \ldots, \chi(X_1))$ forms a Markov chain, hence the complexity of computing $P(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1)$ for any value of x'_r and x'_1 is about $r \times 2^{3m}$ where m is the cipher's block size in byte (i.e. 2^{24} for CS-Cipher, 2^{48} for AES).
Let us detail the computation step :

Data: For a value x'_1 fixed
for $j = 1 \dots r$ **do**
  for $i = 0 \dots 2^m - 1$ **do**
    compute $Pr(\chi(X_j) = i | \chi(X_1) = x'_1)$:
    $\sum_{k=0}^{2^m-1} Pr(\chi(X_j) = i | \chi(X_{j-1}) = k) Pr(\chi(X_{j-1}) = k | \chi(X_1) = x'_1)$
  end
end

Note: From μ properties, we know there is at most $3^4 (< 2^8)$ values of k in the above sum where $Pr(\chi(X_j) = i \mid \chi(X_{j-1}) = k) \neq 0$. Hence the complexity of this computation is, for CSC*, slightly less than 2^{3m}.

2. $Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1)$. Evaluating such a probability is a hard problem in general, therefore we will discuss its approximation.

Due to the fact that the propagation of 0s bytes (i.e. non active S-Boxes) in a differential characteristic is much more predictable than propagations of non-0s bytes values (thanks to the non-linear permutations) we strongly believe that the influence of $\chi(X_1)$ on the value of non-0 bytes of X_r is substantially weaker than its influence on null bytes of X_r. That is to say, the influence of $\chi(X_1)$ on $\chi(X_r)$ is stronger than its influence on X_r given the value of $\chi(X_r)$. Thus, if we assume that

$$Pr(\chi(X_r) = x'_r \mid \chi(X_1) = x'_1) = Pr(\chi(X_r) = x'_r) \pm \epsilon$$

then

$$Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1) = Pr(X_r = x_r \mid \chi(X_r) = x'_r) \pm \epsilon \pm O(\epsilon).$$

Results for CSC:*

– From computation we found that for $r \geq 11$

$$Pr(\chi(X_r) = x'_r) - 2^{-8*m} \leq Pr(\chi(X_r) = x'_r | \chi(X_1) = x'_1) \leq Pr(\chi(X_r) = x'_r) + 2^{-8*m}$$

– We deduce from the above bounds that for $r \geq 11$

$$Pr(X_r = x_r \mid \chi(X_r) = x'_r, \chi(X_1) = x'_1) \begin{cases} \geq Pr(X_r = x_r \mid \chi(X_r) = x'_r) - 2^{-8*m} \\ \leq Pr(X_r = x_r \mid \chi(X_r) = x'_r) + 2^{-8*m} \end{cases}$$

Finally, let us remark that

$$Pr(X_r = x_r \mid \chi(X_r) = x'_r) = (\tfrac{1}{2^8 - 1})^{h(x'_r)}$$
$$Pr(\chi(X_r) = x'_r) = (\tfrac{2^8 - 1}{2^8})^{h(x'_r)} \times (\tfrac{1}{2^8})^{m - h(x'_r)} = (2^8 - 1)^{h(x'_r)} \times 2^{-8*m}$$

And then after 11 rounds (i.e. 4 rounds of CS-Cipher) we have

$$Pr(X_r = x_r \mid X_0 = x_0)$$
$$\begin{cases} \geq ((\frac{1}{2^8-1})^{h(x'_r)} - 2^{-8*m})((2^8-1)^{h(x'_r)}2^{-8*m} - 2^{-8*m}) \times [DP_{min} \times (2^4 - 2^{-4})]^{h(x'_1)} \\ \leq ((\frac{1}{2^8-1})^{h(x'_r)} + 2^{-8*m})((2^8-1)^{h(x'_r)}2^{-8*m} + 2^{-8*m}) \times [DP_{max} \times (2^8 - 1)]^{h(x'_1)} \end{cases}$$

The final bounds of the probability of an r-round differential :

$$\boxed{Pr(X_r = x_r \mid X_0 = x_0) \begin{cases} \geq 2^{-8*m}[DP_{min} \times (2^4 - 2^{-4})]^{h(x'_1)} + O(2^{-8*2*m}) \\ \leq 2^{-8*m}[DP_{max} \times (2^8 - 1)]^{h(x'_1)} + O(2^{-8*m}) \end{cases}}$$

From the above lower bound, we claim that there is no impossible differential on CS-Cipher after 4 rounds and thus CS-Cipher is immune against impossible differential after 6 rounds.

3 Conclusion

Under the strong assumption that CS-Cipher acts very much like a Markov and a Support Markov cipher, we can prove its resistance against impossible differential. To our knowledge this is the first formal result on provable security against impossible differential, even though it remains to be proven that the model is a tight approximation of the cipher.

Future work should focus on this proof and expand the study to other ciphers (particularly AES that has common features with CS-Cipher).

References

1. Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. J. Cryptology 4(1), 3–72 (1991)
2. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 17–38. Springer, Heidelberg (1991)
3. Sugita, M., Kobara, K., Uehara, K., Kubota, S., Imai, H.: Relationships among differential, truncated differential, impossible differential cryptanalyses against word-oriented block ciphers like RIJNDAEL, E2. In: AES Candidate Conference, pp. 242–254 (2000)
4. Stern, J., Vaudenay, S.: Cs-cipher. In: Vaudenay, S. (ed.) FSE 1998. LNCS, vol. 1372, pp. 189–205. Springer, Heidelberg (1998)
5. Vaudenay, S.: On the security of cs-cipher. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 260–274. Springer, Heidelberg (1999)
6. Schnorr, C.P., Vaudenay, S.: Black box cryptanalysis of hash networks based on multipermutations. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 47–57. Springer, Heidelberg (1995)
7. Schnorr, C.P., Vaudenay, S.: La Sécurité des Primitives Cryptographiques (1995)
8. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 12–23. Springer, Heidelberg (1999)

VNSOptClust: A Variable Neighborhood Search Based Approach for Unsupervised Anomaly Detection

Christa Wang and Nabil Belacel *

National Research Council Canada, IIT-Knowledge Discovery Group,
55 Crowley Farm Road, Suite 1100
Moncton, NB E1A 7R1, Canada
{christa.wang,nabil.belacel}@nrc-cnrc.gc.ca

Abstract. In this paper, we present a new algorithm, VNSOptClust, for automatic clustering. The VNSOptClust algorithm exploits the basic Variable Neighborhood Search metaheuristic to allow clustering solutions to get out of local optimality with a poor value; it considers the statistic nature of data distribution to find an optimal solution with no dependency on the initial partition; it utilizes a cluster validity index as an objective function to obtain a compact and well-separated clustering result. As an application for unsupervised Anomaly Detection, our experiments show that (i) VNSOptClust has obtained an average detection rate of 71.2% with an acceptably low false positive rate of 0.9%; (ii) VNSOptClust can detect the majority of unknown attacks from each attack category, especially, it can detect 84% of the DOS attacks. It appears that VNSOptClust is a promising clustering method in automatically detecting unknown intrusions.

Keywords: Unsupervised Learning; Automatic Partitional Clustering; Variable Neighborhood Search; Unsupervised Anomaly Detection.

1 Introduction

Network intrusion attacks pose a serious security threat in a network environment. A wide range of attacks include attempts to destabilize the whole network, to gain unauthorized access to file or privileges, and to prevent legitimate users from using a service [24]. The goals of Intrusion Detection Systems (IDSs) are to automatically detect intrusion attacks from the audit data, and to protect vulnerable network systems with cooperation of static defense mechanisms such as firewalls and software updates [22].

Given the significance of the intrusion detection problems, a number of intrusion detection approaches have been proposed. However, traditional signature-based IDSs suffer from the following drawbacks: first, known signature patterns have to be hand-coded into the systems; secondly, only known attacks that have characteristic signatures can be detected [20]. Data mining

* Corresponding author.

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 607–616, 2008.

based IDSs [4], [5], [16], [19], [20] require precisely labeled data or purely normal data in order to perform misuse detection or anomaly detection [24], [25]. In practice, neither precisely labeled data nor purely normal data is readily available.

To solve the above problems, Portnoy et al. [24] proposed the concept of the unsupervised anomaly detection clustering. The proposed method takes a set of unlabeled data as input and the clustering is performed to separate intrusions and normal instances using a distance-based metric. Once the data is clustered, the normal instances form large clusters while anomalies appear in small clusters. The main advantage of this unsupervised anomaly detection clustering algorithm is the ability to process unlabeled data and automatically detect unknown intrusions.

In this paper, we present a new Variable Neighborhood Search (VNS) based clustering algorithm, VNSOptClust, for solving the unsupervised anomaly detection problem. The VNSOptClust algorithm adopts the basic VNS principle to allow clustering solutions to get out of local optimality to reach a near-global optimum; it considers the statistical nature of data to find a near-globally optimal solution with no dependency on the initial partition status; it utilizes a cluster validity index as an objective function to obtain a compact, well-separated partition. Based on the two assumptions[1] in [24], VNSOptClust can automatically detect intrusion attacks by clustering the unlabeled data and labeling the large clusters as normal and small clusters as abnormal, respectively. The simulation on the subsets of KDD-99 Cup dataset suggests that VNSOptClust is effective in distinguishing anomalies in the dataset from the normal ones. It has obtained an average detection rate of 71.2% with an acceptably low false positive rate of 0.9%. In addition, VNSOptClust can detect the majority of unknown attacks from each attack category. Especially, it can detect 84% of the DOS attacks. Therefore, it appears that VNSOptClust is a promising clustering method in automatically detecting unknown intrusions.

The remainder of the paper is organized as follows: In section 2, the related work in the cluster analysis is reviewed. We confine our discussion on the partitional clustering methods for unsupervised anomaly detection. In Section 3, the VNSOptClust algorithm is introduced in detail. In Section 4, experimental results are reported. Finally, the conclusion is drawn.

2 Related Work in Cluster Analysis

Clustering is a discipline aimed at automatically revealing and describing homogeneous groups or clusters in a dataset. The objective of clustering is that the objects within a group be similar to each other and different from the objects in other groups. In general, the clustering methods can be broadly classified into two categories: hierarchical clustering and partitional clustering. Hierarchical clustering methods build a tree structure for a nested sequence of

[1] Two assumptions: First, the number of normal instances is overwhelmingly larger than the number of intrusions; second, the intrusive instances are qualitatively different from the normal ones.

partitions whereas Partitional clustering methods produce a single partition. In this paper, we will confine our discussion on partitional clustering problems.

The most popular partitional methods are K-means and its variants. K-means is an iterative hill-climbing algorithm and the solution obtained depends on the initial partition status (initial number of clusters with initial centroid seeds). In order to detect the optimal number of initial clusters, an expensive fine-tuning process is necessary. In addition, K-means is often stuck in a local optimum with a poor value and fails to converge to a global optimum. To tackle the shortcomings of K-means, a number of clustering methods have been proposed [8], [11], [24]. The H-Means+ algorithm, an improved version of K-means, eliminates the farthest point that currently contributes most to the total Sum of Squared Errors to improve the clustering performance [11]. In [24], the authors proposed an algorithm for automatic clustering. The algorithm uses a single-linkage clustering, which starts with an empty set of clusters and updates it iteratively. For each data instance, if its distance to the centroid of the selected cluster is less than predefined constant(Cluster Width) then this data instance is assigned to that cluster. Otherwise, a new cluster is created. However, Portnoy's algorithm still requires the proper values of to be predefined manually for each given dataset. To perform automatic clustering without predefining any constants, Guan et al. [8] introduced the Y-means algorithm. Y-means applies postprocessing strategies to adjust the initial number of clusters so that the initial number of clusters and initial centroid seeds are not crucial to the clustering solutions. However, Y-means still belongs to the category of local search heuristics. It often terminates at a local optimum with no guarantee convergence to a global optimum.

To improve the convergence of the clustering performance, several metaheuristic-based optimization methods have been introduced to solve the global optimization problem. The philosophy of such metaheuristic methods is to efficiently explore the search space, to escape from local optima, and to find a near-optimal solution[2]. Among them, Simulated Annealing (SA) [26], Tabu Search (TS) [1], and Genetic Algorithms (GAs) [2], [9], [18], [22], [29] are the commonly-used methods in solving the global optimization problem. However, the main drawbacks of such metaheuristic-based clustering algorithms are parameter selection and high computational complexity [32]. An ideal clustering algorithm should be able to automatically detect near-globally optimal clusters in reasonable time with no dependency on the initial number of clusters and the initial centroid seeds and no need of critical parameter selection.

Variable Neighborhood Search (VNS) is a newly proposed metaheuristic method for solving combinatorial and global optimization problems [12]. The basic principle of VNS is to proceed to a systematic change of neighborhoods within a local search routine. In comparison with other metaheuristics, VNS has several advantages [14]: (i) In VNS, there are no critical parameters to be defined

[2] Since finding the exact global solutions of the clustering problem in a reasonable amount of computational time is an NP-hard problem [27], the goals of solving the global optimization problem are to allow clustering solutions to get out of local optima and to provide near-optimal solutions in reasonable time.

while retaining its efficiency and effectiveness. (ii) VNS can provide near-optimal solutions in moderate computing time.

Inspired by the successful applications of VNS (e.g., Traveling Salesman Problem [13], p- median Problem [10], Minimum Sum-of-Squares Clustering Problem [11], Multi-source Weber Problem [6], [7], and Fuzzy Clustering Problem [3]), we have developed a VNS-based clustering algorithm, VNSOptClust, in automatically searching optimal clusters [31]. In this paper, we will apply VNSOptClust to solve the Unsupervised Anomaly Detection problem.

3 The VNSOptClust Algorithm

VNSOptClust is developed from the basic VNS principle [12]. The basic idea of VNSOptClust is to proceed to a systematic change of neighborhoods within a local search routine. The search is centered around the current best solution and explored increasingly distant neighborhoods until a better solution is found, and then jumped there. VNSOptClust is an optimization process controlled by a random perturbation routine, in which both descend to local optimal and escape from local optima are reached. In this way, VNSOptClust allows clustering solutions to get out of local optima and converge to a near-global optimum. Moreover, VNSOptClust considers the statistical nature of data distribution, eliminating the effect of outliers in clustering procedures, and handling the appearance of empty clusters. Unlike traditional local search methods, VNSOptClust is not sensitive to the initial number of clusters and initial centroid seeds. The general steps of VNSOptClust can be described as follows:

Step 1: *Initialization*
(1) *Assignment*: Partition the normalized data instances ($I_j, j = 1, 2, ..., n$, n is the total number of data instances in the dataset) into p(arbitrary initial number of clusters, $p \in [2, 3, ..., n]$)clusters.
(2) *Remove Empty Clusters*: For each of p clusters, check for empty clusters. If there are, remove them. The resulting number of clusters is p_1.
(3) *Splitting*: For each cluster $C_i, i = 1, 2, ...p_1$, identify outliers based on the splitting condition[3] and replace them as centroids of new clusters.
(4) Let P_M and f_{opt}[4] be the current incumbent partition and the current objective value for VNS heuristic search; choose stopping condition t_{max}(maximum running time for the VNS heuristic search) and a value for the parameter k_{max}(the maximum number of Neighborhoods to be searched).

Step 2: *Termination (Outer Loop)*
If the stopping condition is met, then stop.

[3] Splitting condition: please refer to Section of Splitting for details.
[4] The Objective function: We have employed Dunn's Index, the Davies-Bouldin Index, and Silhouette Validity Index respectively as an objective function and found the clustering results are irrespective with the index being used.

Step 3: *First Neighborhood around current incumbent solution*
Set $k = 1$, k is the current searching neighborhood.

Step 4: *Inner Loop*
If $k > k_{max}$ or $2k > |c|$, where $|c|$ is the number of clusters in the current solution, then return to Step 2 and stop.

Step 5: *Perturbation*
Randomly choose k pairs of clusters from the current solution , and then merge k pairs of clusters into k clusters; denote the so-obtained partition with P_M^1.

Step 6: *Local Search*
(1) *Merging*: With P_M^1 as the initial solution, merge any two clusters in P_M^1 based on the merging condition[5]. The resulting number of clusters is p_2.
(2) *Assignment*: Partition the normalized data instances into p_2 clusters.
(3) Denote the resulting partition and the objective value with P_M^2 and f_{new} respectively.

Step 7: *Move or Not*
If f_{new} is better than f_{opt}, then recenter the search around the new solution P_M^2: Set $f_{opt} = f_{new}$ and $P_M \leftarrow P_M^2$, and go to Step 3. Otherwise, set $k = k + 1$ and go to Step 4.

It should be noted that VNSOptClust does not require any critical parameters to be defined. Since VNSOptClust can automatically detect optimal clusters with no dependency on the initial number of clusters [31], the value of the initial number of clusters is not sensitive to the clustering result. Parameters t_{max}, k_{max} are defined based on the users' expectation of how much time and how far the VNS heuristic search performs. In our experiment, we used $t_{max} = 2$ seconds and $k_{max} = 10$.

As observed, several strategies have been employed in VNSOptClust. VNSOptClust has taken into consideration the statistical nature of data distribution to identify and remove outliers to improve the clustering performance; its effectiveness has been implemented through the procedures of perturbation and local search. We therefore present those strategies in the remainder of the section.

Splitting. The purpose of the splitting procedure is to identify outliers, to remove outliers from each cluster, and to replace them as centroids of new clusters. As the Euclidean distance is used to measure the similarity between any two data points, outliers can be treated as data points that are far from the cluster centroid. VNSOptClust considers the statistical nature of data distribution and applies the Chebyshev's Theorem to determine the splitting threshold.

[5] Merging condition: please refer to Section of Local Search for details.

Chebyshev's Theorem.
For any data distribution, at least $(1 - 1/n^2)$ *of the observations of any set of data lies within* n *deviations of the mean, where* n *is greater than 1.*[30]

By applying Chebyshev's Theorem, we observe that at least 94% of data objects lie within 4 standard deviations of the mean when $n = 4$. It can be assumed that, given majority of data objects (94%) lie within 4 standard deviations of the cluster centroid, the data objects that stay beyond the threshold 4σ can be identified as outliers. Hence, we can define our splitting condition as follows: given the cluster centroid, if any data point within the cluster whose distance from the cluster centroid is greater than the threshold $d = 4\sigma$, then this data point can be identified as an outlier. VNSOptClust removes the identified outlier from the cluster and replaces it as the centroid of a new cluster. The splitting procedure is repeated until no outliers exist.

Perturbation. The objective of the perturbation stage in the VNS heuristic search is to provide a good start for the local search heuristic. To implement the diversification of the VNS heuristic search, the perturbation step randomly selects starting points from the increasingly distant neighborhoods of the current best solution. The process of changing neighborhoods with increasing cardinality in case of no improvements yields a progressive diversification. Perturbation is critical for the VNS heuristic search since choosing random starting points in the neighborhoods of the current best solution is likely to produce a solution that maintains some good features of the current best one.

In VNSOptClust, the local search routine employs the idea of merging two closest clusters. To implement the diversification of the VNS heuristic search, the perturbation step in VNSOptClust randomly select a starting point from the neighborhoods of the current best solution by arbitrarily choosing k pairs of clusters (start with $k = 1$) and merging these k pairs of clusters into k single clusters. If there is no improvement in the VNS heuristic search, VNSOptClust generates a progressive diversification process, in which k is incremented while changing the neighborhoods, and a new perturbation step starts using a different neighborhood.

Local Search. The random solution generated from the procedure of Perturbation becomes the starting point of the local search. To address the issue of dependency on the initial partition status, VNSOptClust applies the Chebyshev's Theorem in the cluster-merging step within the local search routine. According to Chebyshev's Theorem, we observe that when $n = \sqrt{2}$, at least 50% of objects are within 1.414 standard deviations of the mean. Therefore, it can be assumed that, given two adjacent clusters, whose overlap is over the threshold $d = 1.414(\sigma_1 + \sigma_2)$ at least 50% of the data points from these

two adjacent clusters are similar to each other. We can say these two adjacent clusters are close enough to be merged. The merging procedure within the local search routine is to create a compact, well-separated partition. After the merging procedure, VNSOptClust can perform the assignment step to assign data objects to these refined clusters. At the end of the Local Search process, a new partition and new objective value are obtained. The new solution is compared with the current best one and a decision whether to replace the current incumbent solution with the new solution is made during the Move-or-Not stage.

4 Experimental Results

As an application to intrusion detection, VNSOptClust is tested on subsets of the KDD Cup 1999 dataset [17]. We have compared VNSOptClust with one automatic, local search based clustering method (Y-means)[6] [8] and one metaheuristic based clustering method (IDBGC)[7] [22]. The strategy for this comparison study is to create the same experimental environment as mentioned in [22]. Five datasets are exacted from the KDD Cup 1999 dataset. The statistical distribution of attack categories in each dataset is detailed in Fig. 1. Both Y-means and VNSOptClust are coded in Java, and tested on these five datasets. All experiments run on Dell-Intel (R) Pentium (R) M CPU 1.8GHz, 1.00GB of RAM.

To evaluate the performance of the clustering algorithms, we are interested in two indicators: the Detection Rate (DR) and the False Positive Rate (FPR). DR is defined as the number of intrusion instances detected by the algorithm divided by the total number of intrusion instances present in the dataset, whereas FPR equals the number of normal instances incorrectly classified by the algorithm as intrusion divided by the number of normal instances in the dataset [24].

The comparative results of Y-means, VNSOptClust, and IDBGC are displayed in Table 1. VNSOptClust has achieved an average detection rate of 71.2% with a low false positive rate of 0.9%. As noted, the average FPR of VNSOptClust is a bit higher than that of IDBGC, but within an tolerably low value according to the definition in [24]. Hence, we can conclude that VNSOptClust is effective in unsupervised anomaly detection.

The results in Table 2 suggest that under the condition of unsupervised anomaly detection, VNSOptClust is able to detect the majority of unknown attacks for each attack category. In particular, it can detect 84% of the DOS attacks. Therefore, VNSOptClust is effective in automatically detecting unknown intrusion attacks.

[6] Y-means is a good representative of automatic local search based clustering method. In [8], it has been applied for intrusion detection. Its performance has been compared with H-means+, an improved version of K-means. It also has a better intrusion detection rate than Portnoy's algorithm [24].

[7] In the literature, there are not many metaheuristic based clustering algorithms available for solving intrusion detection problems. The best solution of intrusion detection based metaheuristic algorithms is taken from [22].

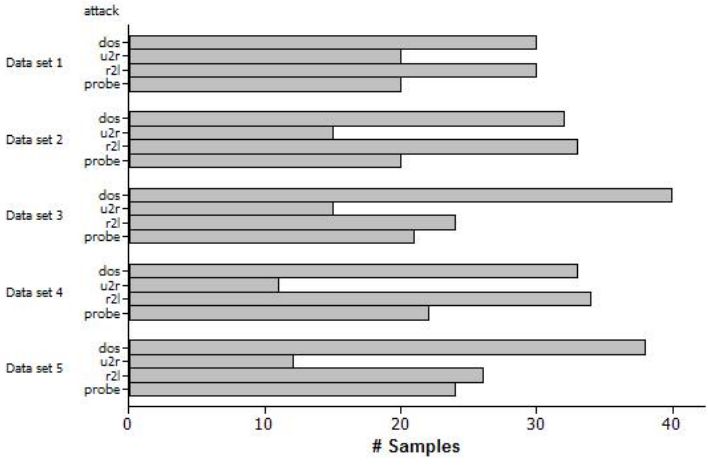


Fig. 1. Attack Distribution in Each Dataset

Table 1. Comparative Results of Y-means, VNSOptClust, and IDBGC

Data set (%)	Y-means DR	Y-means FPR	VNSOptClust DR	VNSOptClust FPR	IDBGC DR	IDBGC FPR
Dataset 1	64	2.0	63	0.4	68	0.8
Dataset 2	55	2.1	81	1.9	33	0.2
Dataset 3	56	1.7	79	0.6	74	0.4
Dataset 4	59	2.2	72	1.0	44	0.3
Dataset 5	52	1.8	64	0.2	79	0.4
Average	57.2	1.96	71.2	0.9	59.6	0.4

Table 2. Detection Percentage of Different Attack Categories

Data set (%)	DOS DR	U2R DR	R2L DR	PROBE DR
Dataset 1	80	70	55	33
Dataset 2	78	80	78	90
Dataset 3	95	67	67	71
Dataset 4	87	72	69	50
Dataset 5	80	50	73	36
Average	84	68	68	56

5 Conclusion

In this paper, we applied a VNS-based clustering algorithm, VNSOptClust, in solving the unsupervised anomaly detection problem. VNSOptClust adopts a VNS metaheuristic procedure to allow clustering solutions to get out of local

optimality with a poor value; it considers the statistical nature of data distribution to find a near-optimal solution; it utilizes a cluster validity index as an objective function of the VNS heuristic search to obtain compact, well-separated clusters. Under the condition of unsupervised anomaly detection, VNSOptClust has obtained an average detection rate of 71.2% with an acceptably low false positive rate of 0.9%, and is capable of detecting the majority of unknown attacks for each attack category. Therefore, VNSOptClust is a promising clustering method for unsupervised anomaly detection.

Acknowledgement. This work was partially supported by NSERC discovery grants awarded to Dr. Nabil Belacel.

References

1. Al-Sultan, K.S.: A Tabu Search Approach to the Clustering Problem. Pattern Recognition 28(9), 1443–1451 (1995)
2. Babu, G.P., Hall, D.: A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm. Pattern Recognition Letters 14, 763–769 (1993)
3. Belacel, N., Hansen, P., Mladenovic, N.: Fuzzy J-means: a new heuristic for fuzzy clustering. Pattern Recognition 35(10), 2193–2200 (2002)
4. Bloedorn, E., Christiansen, A.D., Hill, W., Skorupka, C., Talbot, L.M., Tivel, J.: Data Mining for Network Intrusion Detection: How to Get Started. MITRE Technical Paper (2001)
5. Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., Tan, P.N.: Data Mining for Network Intrusion Detection. In: Proceedings of the NSF Workshop on Next Generation Data Mining, Baltimore, Maryland (2002)
6. Brimberg, J., Mladenovic, N.: A variable neighborhood algorithm for solving the continuous location-allocation problem. Studies in Location Analysis 10, 1–12 (1996)
7. Brimberg, J., Hansen, P., Mladenovic, N., Tailard, E.: Improvements and Comparison of heuristics for solving the multisource weber problem. Operations Research 3, 444–460 (2000)
8. Guan, Y., Ghorbani, A., Belacel, N.: Y-means: a clustering method for intrusion detection. In: Proceedings of 2003 IEEE Canadian Conference on Electrical and Computer Engineering, Montreal, pp. 1083–1086 (2003)
9. Hall, L.O., Ozyurt, I.B., Bezdeck, J.C.: Clustering with a Genetically Optimized Approach. IEEE Transaction on Evolutionary Computation 3(2), 103–112 (1999)
10. Hansen, P., Jaumard, B., Mladenovic, N., Parreira, A.: Variable Neighborhood Search for the - median Location Science 5(4), 207–226 (1998)
11. Hansen, P., Mladenovic, N.: J-means: a new local search heuristic for minimum sum of squares clustering. Pattern Recognition 34, 405–413 (2001)
12. Hansen, P., Mladenovic, N.: Variable Neighborhood Search. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 145–184. Kluwer Academic Publishers, Boston (2003)
13. Hansen, P., Mladenovic, N.: Variable Neighborhood Search: Principle and Applications. European Journal of Operational Research 34, 405–413 (2001)

14. Hansen, P., Mladenovic, N.: An Introduction to Variable Neighborhood Search. In: Vo, S., Martello, S., Osman, I., Roucairol, C. (eds.) Metaheuristics: Advances and trends in local search paradigms for optimization, pp. 433–458. Kluwer Academic Publishers, Dordrecht (1999)
15. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, New Jersey (1988)
16. Julisch, K.: Data Mining for Intrusion Detection: A Critical Review. In: Barbara, D., Jajodia, S. (eds.) Applications of data mining in computer security, pp. 1–14. Kluwer Academic Publisher, Boston (2002)
17. KDD Cup 1999 Data, University of California, Irvine (October 1999), http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
18. Krishna, K., Murty, M.: Genetic K-means Algorithm. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 29(3), 433–439 (1999)
19. Lee, W., Stolfo, S.J., Chan, P.K., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J.: Real time data mining based intrusion detection. In: Proceedings of DARPA Information Survivability Conference & Exposition II (2001)
20. Lee, W., Stolfo, S.J.: Data Mining Approaches for Intrusion Detection. In: Proceedings of the 7th USEUIX Security Symposium, San Antonio, Texas (1998)
21. Lippmann, R.P., Graf, I., et al.: The 1998 DARPA/AFRL Off-Line Intrusion Detection Evaluation. In: First International Workshop on Recent Advances in Intrusion Detection (RAID), ouvain-la-Neuve, Belgium (1998)
22. Liu, Y., Chen, X., Liao, X., Zhang, W.: A genetic clustering method for intrusion detection. Pattern Recognition 37, 927–942 (2004)
23. Mirkin, B.: Clustering for Data Mining: A Data Discovery Approach. CRC Press, Boca Raton (2005)
24. Portnoy, L., Eskin, E., Stolfo, S.J.: Intrusion Detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), Philadelphia, PA (2001)
25. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65 (1987)
26. Selim, S.Z., Alsultan, K.: A Simulated Annealing Algorithm for the Clustering Problem. Pattern Recognition 24(10), 1003–1008 (1991)
27. Spath, H.: Cluster Analysis Algorithms. Ellis Horward, Chichester (1980)
28. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining, pp. 487–647. Addison-Wesley, Boston (2006)
29. Tseng, L.Y., Yang, S.B.: A genetic approach to the automatic clustering problem. Pattern Recognition 34, 415–424 (2001)
30. Walpole, R.E.: Elementary Statistical Concepts, 2nd edn. Macmillan Publishing Co., New York (1983)
31. Wang, C. In: Search of Optimal Clusters Using Variable Neighbourhood Search, Master Thesis, Department of Computer Science, University of New Brunswick, Fredericton, Canada (2007)
32. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)

Author Index