Peter J. Bentley
Doheon Lee
Sungwon Jung (Eds.)

LNCS 5132

Artificial Immune Systems

7th International Conference, ICARIS 2008
Phuket, Thailand, August 2008
Proceedings

Springer

Lecture Notes in Computer Science 5132

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Peter J. Bentley Doheon Lee
Sungwon Jung (Eds.)

Artificial Immune Systems

7th International Conference, ICARIS 2008
Phuket, Thailand, August 10-13, 2008
Proceedings

Springer

Volume Editors

Peter J. Bentley
University College London, Department of Computer Science
Malet Place, London WC1E 6BT, UK
E-mail: p.bentley@cs.ucl.ac.uk

Doheon Lee
Sungwon Jung
Korea Advanced Institute of Science and Technology
IBM-KAIST Bio-Computing Research Center
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea
E-mail: dhlee@kaist.ac.kr, sjung@tgen.org

Library of Congress Control Number: 2008931996

CR Subject Classification (1998): F.1, I.2, F.2, H.2.8, H.3, J.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-85071-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-85071-7 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12442546 06/3180 5 4 3 2 1 0

7th International Conference on Artificial Immune Systems

10th–13th August, 2008 in Phuket, Thailand

There are many desirable features of natural systems: adaptability, robustness, homeostasis, memory, immunity. Biological immune systems seem to exhibit all of these features and more. Thus it is not so surprising that a vigorous research field has emerged, which focuses on understanding biological immune systems and creating new models, algorithms, technologies and theoretical understandings. The field is known collectively as artificial immune systems (AIS), and comprises a remarkably diverse range of researchers. Biologists join forces with mathematicians to create new models. Engineers and computer scientists produce new autonomous intelligent software. Roboticists and specialists in unconventional computation create new control systems or new ways to compute.

The International Conference on Artificial Immune Systems is proud to be the premiere conference in this exciting area. For the first time ICARIS moved to East Asia, not only being held in Thailand with Thai local chairs, but also with conference chairs from South Korea. As its organizers, we were honored to have had such a variety of innovative and original scientific papers presented this year, especially from those new to the conference.

ICARIS 2008 was the seventh international conference dedicated entirely to the field of AIS. We had more submissions than ever before this year, and because our acceptance rate is based purely on quality, we accepted 60% of papers. These acceptances were based on advice from stream leaders – experts in the field who agreed to help monitor submissions and make decisions on subject and quality. Thus, in these proceedings you will find 40 papers written by the leading scientists in the field, from 25 different countries in 4 continents, describing an impressive array of ideas, technologies and applications for AIS. We could not have organized this conference without these researchers, so we thank them all for coming. We also could not have organized ICARIS without the excellent work of all of the Programme Committee, our Publicity Chair Sungwon Jung, our Local Chairs Supiya Charoensiriwath and Boonserm Kaewkamnerdpong, and our conference administrator, J.J. Giwa.

Whether you are new to the field, or are one of its established researchers, we hope you enjoy the proceedings of ICARIS 2008.

June 2008

Doheon Lee
Peter J. Bentley

Organizing Committee

Conference Chairs

Doheon Lee — KAIST, Korea, dhlee@biosoft.kaist.ac.kr
Peter Bentley — University College, London. UK, P.Bentley@cs.ucl.ac.uk

Local Conference Chairs

Supiya Ujjin — NECTEC, Thailand, ujjins@gmail.com
Boonserm Kaewkamnerdpong — NECTEC, Thailand, boonserm@gmail.com

Publicity Chair

Sungwon Jung — KAIST, Korea, swjung@biosoft.kaist.ac.kr

Stream Leaders

1. Computational Immunology
Emma Hart — Napier University, UK, E.Hart@napier.ac.uk
2. Applied AIS
Henry Lau — Hong Kong University, China, hyklau@hkucc.hku.hk
Vincenzo Cutello — Catania University, Italy, cutello@dmi.unict.it
3. Theoretical AIS
Andy Hone — Kent University, UK, A.N.W.Hone@kent.ac.uk
4. Position papers
Jon Timmis — York University, UK, jtimmis@cs.york.ac.uk
Paul Andrews — York University, UK, psa@cs.york.ac.uk

Conference Administrator

J.J. Giwa

Keynote Speakers

Norman Packard (Santa Fe Institute / ProtoLife)
Julie McLeod (University of West of England)

Tutorial Speakers

1. Introduction to Artificial Immune Systems
Giuseppe Nicosia (University of Catania)
2. Applied Artificial Immune Systems
Leandro De Castro (Catholic University of Santos)
3. Simulating and Modelling the Immune System
Part A: Effective Use of OO Techniques for Easy Simulation of Immune Systems: A Toolkit for the Immunologist: Hugues Bersini, (ULB)
Part B: Understanding the Immune System Through Modelling and Simulation: A Toolkit for the Engineer: Emma Hart (Napier University)

Programme Committee

Alex Freitas	University of Kent, UK
Alexander Tarakanov	St. Petersburg Inst. for Info. and Auto., Russia
Andrew Watkins	University of Kent, UK
Andy Hone	University of Kent, UK
Andy Tyrrell	University of York, UK
Carlos A. Coello Coello	CINVESTAV-IPN, Mexico
Carlos Fernando Esponda Darlington	Yale University, USA
Christian Jacob	University of Calgary, Canada
Colin Johnson	University of Kent, UK
Dipankar Dasgupta	University of Memphis, USA
Doheon Lee	KAIST, Korea
Emma Hart	Napier University, UK
Ernesto Costa	University de Coimbra, Portugal
Fabio Gonzalez	National University of Colombia, Colombia
Fernando J. Von Zuben	State University of Campinas, Brazil
Giuseppe Nicosia	University of Catania, Italy
Henry Lau	University of Hong Kong, China
Hugues Bersini	IRIDIA, ULB, Belgium
Jon Timmis	University of York, UK
Julie Greensmith	University of Nottingham, UK
Leandro de Castro	LSIn, UniSantos, Brazil
Licheng Jiao	Xidian University, China
Luis Fernando Nino	Nat'l University of Colombia, Colombia
Maoguo Gong	Xidian University, China
Mark Neal	University of Wales, Aberystwyth, UK
Mario Pavone	University of Catania, Italy
Myriam R.B.S. Delgado	CEFET-PR, Brazil
Nikolaos Nanas	Ctr. for Research and Technology (CERETETH), Greece
Paul Andrews	University of York, UK
Peter Bentley	UCL, UK
Peter Ross	Napier University, UK

Simon Garrett	University of Wales, UK
Siti Zaiton Mohd Hashim	University Technology Malaysia, Malaysia
Slawomir T. Wierzchon	Polish Academy of Sciences, Poland
Stephanie Forrest	University of New Mexico, USA
Steve Cayzer	Hewlett Packard Laboratories, UK
Susan Stepney	University of York, UK
Thomas Stibor	Darmstadt University of Technology, Germany
Uwe Aickelin	University of Nottingham, UK
Vincenzo Cutello	University of Catania, Italy
Wenjian Luo	University of Science and Technology of China, China
Ying Tang	Peking University, China

Table of Contents

Computational Immunology

Applied AIS

Theoretical AIS

Position/Conceptual Papers

A Stochastic Model of the Interleukin (IL)-1β Network

Johnny Kelsey[1], Brian Henderson[2], Rob Seymour[3], and Andy Hone[4]

[1] CoMPLEX, University College London
[2] Division of Microbial Diseases, University College London
[3] CoMPLEX/Department of Mathematics, University College London
[4] IMSAS, University of Kent

Abstract. The interleukin-1β network is a primary mediator of the inflammatory response and plays an important role in many immunological processes. A Markov chain model of the network is presented, along with results from iteration over the stochastic matrix. The stationary distribution of the model is analysed.

Keywords: IL-1β; interleukin-1β; Markov process; cytokine network; stochastic matrix.

1 Introduction

Interleukin-1β (IL-1β) is a cytokine, a polypeptide mediator used by the immune system to communicate between cells. IL-1 has been described as the most potent and multifunctional cell activator in immunology and cell biology [8]; it plays many essential roles in the immune system. The complexity of the IL-1 network has been noted by many researchers [3], [10]. IL-1 has two forms, IL-1α and IL-1β. There are two receptors which bind IL-1. A receptor accessory protein is necessary to form a signalling complex; many inhibitory factors are part of the network. We shall examine a subset of the IL-1β network using stochastic techniques to find out whether or not a stationary distribution exists over signalling and nonsignalling states.

2 The IL-1β Network

We shall focus on the IL-1β form of IL-1. IL-1β binds to two receptors on the cell membrane, the type-I and type-II receptor: the type-I receptor can cause a signal transduction event; the type-II receptor is a *decoy receptor*, lacking the transmembrane apparatus to initiate a signalling event [8].

When IL-1β binds to the type-I receptor, a *signalling binary complex* is formed. Signal transduction does not occur, however, until a *receptor accessory protein* binds to the signalling binary complex, forming a *signalling ternary complex*.

The type-II receptor can also bind to IL-1β to form a *nonsignalling binary complex*. The nonsignalling binary complex can also bind the receptor accessory

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 1–11, 2008.

protein to form a *nonsignalling ternary complex*. Thus, the type-II receptor competes both for IL-1β and the receptor accessory protein; it is a key inhibitory component in the network.

The type-I receptor is not abundant, but evokes a powerful response without a high level of receptor occupancy [2], as the receptor activates many pathways which operate in parallel. Unlike most other cytokines, it is thought that as few as ten occupied receptors are sufficient to evoke a strong response [20]. Since IL-1β typically acts at very low concentrations, the population sizes of signalling and nonsignalling complexes will be small, and random fluctuations will have a disproportionate effect. The use of stochastic methods is indicated to model such a system.

The interactions we model in this paper are as follows:

- IL-1β + type-I receptor: IL-1β associates with the signalling receptor to form a signalling binary complex
- signalling binary complex + receptor accessory protein: promotion of the receptor accessory forms a signalling ternary complex, and signalling occurs
- IL-1β + type-II receptor: IL-1β associates with the nonsignalling receptor to form a nonsignalling binary complex
- nonsignalling binary complex + receptor accessory protein: promotion of the receptor accessory forms a nonsignalling ternary complex

As in any reaction, the binding event is reversible; the complexes can both associate and dissociate. The association and dissociation rates for the binary and ternary signalling and nonsignalling complexes are given in Table (1).

We use the following notation: L is the free (unbound) IL-1β, $R1$ is the type-I signalling receptor, $R2$ is the type-II nonsignalling receptor, S is the signalling binary complex, NS is the nonsignalling binary complex, R is the receptor accessory protein, T is the signalling ternary complex, NT is the nonsignalling ternary complex. Association rates are given as k_u^+ for a arbitrary component u, and dissociation rates are k_u^-. In the notation of chemical reactions, the interactions are:

$$L + R1 \underset{k_S^-}{\overset{k_S^+}{\rightleftharpoons}} S \tag{1}$$

$$L + R2 \underset{k_{NS}^-}{\overset{k_{NS}^+}{\rightleftharpoons}} NS \tag{2}$$

$$S + R \underset{k_T^-}{\overset{k_T^+}{\rightleftharpoons}} T \tag{3}$$

$$NS + R \underset{k_{NT}^-}{\overset{k_{NT}^+}{\rightleftharpoons}} NT \tag{4}$$

Units for dissociation rates k_u^- are s^{-1}, and these rates can be construed as probabilities per unit time (= 1s). However, the units for association rates, k_u^+, are $\mathrm{M}^{-1}\mathrm{s}^{-1}$, and these cannot be interpreted as probabilities without a transformation, $k_u^+ = c\hat{k}_u^+$, where c is a suitable conversion factor having dimension M (concentration).

We choose c to be a concentration based on one international unit of specific activity of IL-1β. A standard international unit (IU) of IL-1β activity is defined as a preparation (NISBC code: 86/632) which contains 0.75mg per ampoule with assigned potency of 75000 units per ampoule [18]. This can be expressed as a standardised concentration:

$$\frac{\text{Unit}}{\text{ml}} = 5.9 \times 10^{-11}\ \text{M}$$

We use this as our conversion factor c to express association rates as probabilities. The derived probabilities are given in Table (2).

Table 1. Association/dissociation rates: the association and dissociation rates of the signalling and nonsignalling complexes of the IL-1β network, where S is the signalling binary and NS the nonsignalling binary

Agent	Binds	Assoc. ($M^{-1}s^{-1}$)	Dissoc. (s^{-1})
IL-1RI [1]	IL-1β	4.67×10^7	1.6×10^{-11}
S [1]	IL-1RAcP	4.67×10^7	0.32×10^{-11}
IL-1RII [2]	IL-1β	8.85×10^4	6.92×10^{-10}
NS [2]	IL-1RAcP	9.5×10^4	6.82×10^{-10}

[1] Source: [15] [2] Source: [21]

3 Markov Chain Model

The theory of discrete-time Markov chains provides powerful techniques for modelling random processes which are generally straightforward to implement computationally [13].

A Markov chain can be described by a diagram showing the transitions between the various states of a system. Figure (1) shows the transition graph for the subset of the IL-1β network directly associated with receptor binding, both signalling and nonsignalling.

The vertices of the diagram represent possible states which a unit of IL-1β can occupy. The arrows represent directed state transitions, with their associated (non-zero) probabilities. A self-loop at a vertex represents the probability that the current state does not change in a given time step.

Notice that the diagram has excluded both the receptors and the receptor accessory protein; we will assume that sufficient resources of these components exist to form signalling and nonsignalling complexes, since the components we wish to model are the unbound IL-1β and the complexes themselves. Estimates of the number of type-I and type-II receptors and accessory proteins R on human

Table 2. Probabilities $\mathbb{P}(X)$ such that $\hat{k}_u^{\pm} \in [0,1]$ derived from association and dissociation rates $k_u^{\pm}$

Association probabilities		
Complex	Symbol	$\mathbb{P}(X)$
Signalling binary	$\hat{k}_S^+$	2.75×10^{-3}
Nonsignalling binary	$\hat{k}_{NS}^+$	5.22×10^{-8}
Signalling ternary	$\hat{k}_T^+$	2.75×10^{-3}
Nonsignalling ternary	$\hat{k}_{NT}^+$	6.5×10^{-8}
Dissociation probabilities		
Complex	Symbol	$\mathbb{P}(X)$
Signalling binary	$\hat{k}_S^-$	1.6×10^{-11}
Nonsignalling binary	$\hat{k}_{NS}^-$	6.92×10^{-10}
Signalling ternary	$\hat{k}_T^-$	0.32×10^{-11}
Nonsignalling ternary	$\hat{k}_{NT}^-$	6.82×10^{-10}

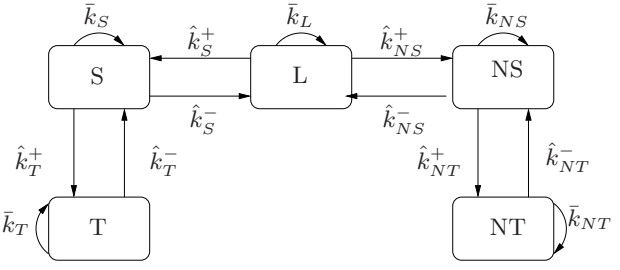


Fig. 1. State transition diagram for IL-1β network, where L is IL-1β, S is the signalling binary complex, NS is the nonsignalling binary complex, T is the signalling ternary complex, and NT the nonsignalling ternary complex. Probabilities are derived from the association and dissociation rates $\hat{k}_u^{\pm}$.;probabilities for self-interaction loops are given by $\bar{k}_i = 1 - \sum_j p_{ij},\ j \neq i$.

and murine cells yield a range of $200 \leq R \leq 2000$ [16], [4], [5], [6], [19]. Since, as already noted, the number of signalling ternary complexes required to initiate signal transduction is relatively low in comparison to the number of receptors expressed on the membrane, the exclusion of receptors and accessory proteins from the model would appear to be justified.

The matrix of transition probabilities defining the Markov process illustrated in Figure (1) is given in equation (5).

$$\mathbf{P} = \begin{bmatrix} & \text{IL}-1\beta & \text{S} & \text{NS} & \text{T} & \text{NT} \\ \text{IL}-1\beta & \bar{k}_L & \hat{k}_S^+ & \hat{k}_{NS}^+ & 0 & 0 \\ \text{S} & \hat{k}_S^- & \bar{k}_S & 0 & \hat{k}_T^+ & 0 \\ \text{NS} & \hat{k}_{NS}^- & 0 & \bar{k}_{NS} & 0 & \hat{k}_{NT}^+ \\ \text{T} & 0 & \hat{k}_T^- & 0 & \bar{k}_T & 0 \\ \text{NT} & 0 & 0 & \hat{k}_{NT}^- & 0 & \bar{k}_{NT} \end{bmatrix} \tag{5}$$

Table 3. The results of iterating over the stochastic matrix P. The labels are: L, unbound ligand; S, signalling binary; NS, nonsignalling binary; T, signalling ternary; NT, nonsignalling ternary. The experiments consisted of n iterations as given in the rightmost column, and each experiment ran $N = 500$ trials. The number of times the systems is in a state S is recorded for each experiment and then averaged by the number of iterations n and repetitions of the experiment N. As can be seen, the amount of time the system spent in the signalling ternary state increases with the number of iterations.

L	S	NS	T	NT	Iterations
0.18756	0.18788	0.20154	0.23902	0.18400	100
0.13538	0.20607	0.20800	0.26855	0.18200	200
0.11487	0.17557	0.18178	0.32176	0.20600	500
0.07669	0.11322	0.20676	0.39731	0.20600	1000
0.03934	0.08522	0.17392	0.52157	0.17993	2000
0.01902	0.03195	0.15765	0.58935	0.20200	5000
0.00954	0.01809	0.14662	0.63773	0.18800	10000
0.00608	0.01085	0.10482	0.66330	0.21493	20000
0.00258	0.00464	0.05216	0.73181	0.20878	50000
0.00168	0.00215	0.02924	0.75901	0.20790	100000
0.00069	0.00111	0.01497	0.77371	0.20949	200000
0.00033	0.00047	0.00775	0.80312	0.18831	500000
0.00018	0.00028	0.00432	0.87185	0.12335	1000000
0.00008	0.00014	0.00276	0.89592	0.10108	2000000
0.00003	0.00005	0.00116	0.95572	0.04301	5000000
0.00002	0.00003	0.00061	0.96736	0.03196	1×10^7
0.00001	0.00001	0.00031	0.98529	0.01437	2×10^7
4.08×10^{-6}	6.10×10^{-6}	0.00012	0.99430	0.00555	5×10^7
2.65×10^{-6}	2.99×10^{-6}	0.00006	0.99757	0.00236	1×10^8
1.13×10^{-6}	1.59×10^{-6}	0.00002	0.99824	0.00172	2×10^8
4.37×10^{-7}	6.67×10^{-7}	9.94×10^{-6}	0.99942	0.00056	5×10^8

We experimented computationally to investigate the amount of time the stochastic matrix P spends in each state, and to find any stationary distributions to which the Markov process is attracted in the long run. Since the process is stochastic, it will not always take the same amount of time to reach an equilibrium state (if any such state exists), which necessitated a large number of experimental trials.

Table (3) shows the results of repeating the experiment and averaging it over the number of iterations n and the number of trials N. Each trial iterates over the stochastic matrix for n iterations, where $100 \leq n \leq 5 \times 10^8$; each experiment was repeated $N = 500$ times. The initial state is set randomly, and then evolves according to the probabilities based on association and dissociation rates. After the experiments have run, we calculate the average of how long the system spent in any particular state.

As can be seen from the table, with a low number of iterations the Markov chain is evenly distributed between its states. However, as the number of iterations increases, the stochastic matrix is rapidly attracted to the signalling ternary state, and spends an increasing number of iterations in this state. It

would seem that the signalling ternary state is, in dynamical systems terms, a robust attractor for the stochastic matrix.

4 Effect of Inhibitory Receptor

We would like to know how long it takes before the matrix approaches equilibrium. To find the time taken to reach equilibrium we iterated over the stochastic matrix until it reached the signalling ternary state, and recorded the number of iterations it took; that is, recorded the length of the Markov chain.

The protocol for the experiment consisted of creating a random initial state, and then iterating over the stochastic matrix P, creating a list of the states for each iteration until it reaches the signalling ternary state. Each experimental trial is a sample path of the Markov chain. The number of iterations is recorded as the result of the trial. The experiment was run for a large number of trials to show the dynamics of the system.

We can observe the behaviour of the system when there is no inhibitory, type-II nonsignalling receptor. Repeating the experiment without the nonsignalling binary (NS) and ternary (NT) states shows what would happen if the network consisted only of the type-I signalling receptor. Thus we can directly compare the behaviour of the sample paths in the presence or absence of the inhibitory receptor, in order to illustrate its effect.

The results from both experiments are given in Figure (2). The experiment consisted of $N = 5000000$ trials. The maximum possible number of iterations over the matrix was set to $n = 50000$; if the matrix arrived at the signalling ternary complex before n, the program terminated and the number of iterations recorded. The

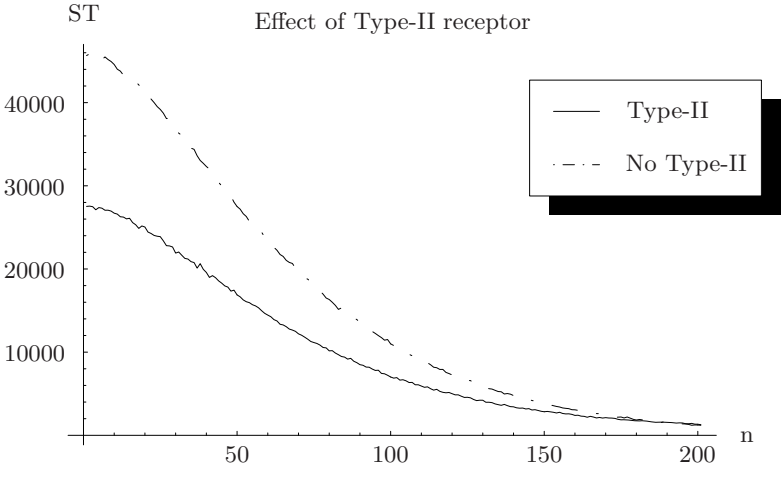


Fig. 2. Sample paths: n are the number of iterations taken for the stochastic matrix to reach the signalling ternary state, ST the number of signalling ternary complexes formed. Each experiment consisted of $N = 5 \times 10^6$ trials. The type-II receptor slows the formation of signalling ternary complexes.

Table 4. Mean path length and standard deviation from experiments: effect of type-II receptor

Experiment	Mean	Standard deviation
Type-II receptor present	3713.96	8154.79
Type-II receptor absent	363.96	468.49

effect of type-II receptors is to slow the formation of signalling ternary complexes; without the type-II receptor, the complexes form much more quickly.

The means and standard deviations for both experiments are given in Table (4). Notice that the mean Markov chain length of the experiment with type-II receptors is an order of magnitude higher than the mean of the experiment without type-II receptors. The sample paths take on average an order of magnitude longer to reach the signalling ternary complex state with the type-II receptor present. From these experiments, we can observe that the type-II nonsignalling receptor slows the formation of signalling ternary complexes, thus indirectly slowing the response of the cell to IL-1β.

5 Long-Term Behaviour of the System

The transition matrix $\mathbf{P}$ given in equation (5) depends on only eight parameters, namely $\hat{k}_S^{\pm}, \hat{k}_{NS}^{\pm}, \hat{k}_T^{\pm}, \hat{k}_{NT}^{\pm}$, because the diagonal entries are determined in terms of these parameters by the requirement that the sum of the entries in each row must be 1 (since P is a stochastic matrix). From Figure (11), we see that it is possible to move from any state to any other along a path having positive probability. That is, the process defined by $\mathbf{P}$ is ergodic.

Standard theorems [13] tell us that, for an ergodic process, a stationary distribution π exists and also satisfies $\pi_{\mathbf{j}} > 0$.

Let

$$\begin{aligned}\psi = \hat{k}_{NS}^{+}\hat{k}_{NT}^{+}\hat{k}_S^{-}\hat{k}_T^{-} + \hat{k}_S^{+}\hat{k}_T^{+}\hat{k}_{NS}^{-}\hat{k}_{NT}^{-} + \hat{k}_{NS}^{+}\hat{k}_S^{-}\hat{k}_T^{-}\hat{k}_{NT}^{-} \\ +\hat{k}_S^{+}\hat{k}_{NS}^{-}\hat{k}_T^{-}\hat{k}_{NT}^{-} + \hat{k}_S^{-}\hat{k}_{NS}^{-}\hat{k}_T^{-}\hat{k}_{NT}^{-}\end{aligned} \tag{6}$$

Solving the equation $\pi\mathbf{P} = \pi$ algebraically, we find:

$$\begin{aligned}\pi_1 &= \frac{\hat{k}_S^{-}\hat{k}_{NS}^{-}\hat{k}_T^{-}\hat{k}_{NT}^{-}}{\psi} \\ \pi_2 &= \frac{\hat{k}_S^{+}\hat{k}_{NS}^{-}\hat{k}_T^{-}\hat{k}_{NT}^{-}}{\psi} \\ \pi_3 &= \frac{\hat{k}_{NS}^{+}\hat{k}_S^{-}\hat{k}_T^{-}\hat{k}_{NT}^{-}}{\psi} \\ \pi_4 &= \frac{\hat{k}_S^{+}\hat{k}_T^{+}\hat{k}_{NS}^{-}\hat{k}_{NT}^{-}}{\psi} \\ \pi_5 &= \frac{\hat{k}_{NS}^{+}\hat{k}_{NT}^{+}\hat{k}_S^{-}\hat{k}_T^{-}}{\psi}\end{aligned} \tag{7}$$

Substituting numerical values from Table (2) gives:

$$\begin{aligned}
\pi_1 &\approx 6.77 \times 10^{-14} \\
\pi_2 &\approx 1.16 \times 10^{-7} \\
\pi_3 &\approx 5.11 \times 10^{-17} \\
\pi_4 &\approx 1 \\
\pi_5 &\approx 4.86 \times 10^{-18}
\end{aligned} \tag{8}$$

Clearly, the values of π_1, π_2, π_3 and π_5 are many orders of magnitude smaller than π_4, which is extremely close to 1. Hence, to all intents and purposes, the system ends up in the signalling ternary state (with probability 1), irrespective of the initial distribution ($\pi_{\mathbf{0}}$).

Given that the probabilities which represent the dissociation of the complexes, namely $\hat{k}_S^-, \hat{k}_{NS}^-, \hat{k}_T^-, \hat{k}_{NT}^-$, are so small, it may be instructive to consider the limiting case when all of these parameters are set to zero. In that case, the transition diagram for the Markov chain can be represented schematically by

$$T \longleftarrow S \longleftarrow L \longrightarrow NS \longrightarrow NT, \tag{9}$$

from which it is clear that the chain is reducible in this special case. Indeed, if the system leaves any state other than one of the ternary complexes (T or NT) then it can never return there. The transition matrix for this special case has the upper triangular form

$$\mathbf{P} = \begin{pmatrix}
1 - \hat{k}_S^+ - \hat{k}_{NS}^+ & \hat{k}_S^+ & \hat{k}_{NS}^+ & 0 & 0 \\
0 & 1 - \hat{k}_T^+ & 0 & \hat{k}_T^+ & 0 \\
0 & 0 & 1 - \hat{k}_{NT}^+ & 0 & \hat{k}_{NT}^+ \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}. \tag{10}$$

We can again solve the vector equation $\pi\mathbf{P} = \pi$ for the reduced matrix (10), which results in a family of stationary distributions

$$\begin{aligned}
\pi_1 &= 0 \\
\pi_2 &= 0 \\
\pi_3 &= 0 \\
\pi_4 &= q \\
\pi_5 &= 1 - q
\end{aligned} \tag{11}$$

Compare this with the stationary distribution we found where the dissociation probabilities are non-zero, given in (8); there, all states would tend toward the signalling ternary complex state T with a probability $\pi_4 \approx 1$. In this case, with the dissociation probabilities set to zero, we have a one-parameter family of stationary distributions, with the parameter q such that $0 \leq q \leq 1$. There are

therefore two possible outcomes for the system, the signalling and nonsignalling ternary states. The non-uniqueness of the stationary distribution means that that the limiting state of the chain is highly dependent on the initial distribution.

What are the probabilities of reaching either state? Firstly, we can see that if the system has the initial state $X_0 = S$, then it can only remain in state S and then (after a finite number of steps, m say) transit to state T, where it will then remain, and so $P(X_\infty = T \,|\, X_0 = S) = \sum_{m=0}^{\infty} \hat{k}_T^+ (1 - \hat{k}_T^+)^m = 1$. Similarly $P(X_\infty = NT \,|\, X_0 = NS) = 1$.

On the other hand, for the initial state $X_0 = L$, at each step the system can stay in that state with probability $(1 - \hat{k}_S^+ - \hat{k}_{NS}^+)$, or transit to S with probability $\hat{k}_S^+$ (in which case it will ultimately reach T with probability 1), or transit to NS with probability $\hat{k}_{NS}^+$ (in which case it will ultimately reach NT).

Summing over transitions to S after m steps, for each m, gives $P(X_\infty = T \,|\, X_0 = L) = \sum_{m=0}^{\infty} \hat{k}_S^+ (1 - \hat{k}_S^+ - \hat{k}_{NS}^+)^m$, and an analogous formula holds for $P(X_\infty = NT \,|\, X_0 = L)$. Hence we see that if the system starts off with an unbound ligand, so $X_0 = L$, then it can end up in either of the ternary states, with the limiting probabilities being

$$P(X_\infty = T \,|\, X_0 = L) = \frac{\hat{k}_S^+}{(\hat{k}_S^+ + \hat{k}_{NS}^+)} = q, \tag{12}$$

$$P(X_\infty = NT \,|\, X_0 = L) = \frac{\hat{k}_{NS}^+}{(\hat{k}_S^+ + \hat{k}_{NS}^+)} = 1 - q. \tag{13}$$

What observations can we make from this analysis? It seems that the dissociation probabilities, despite their insignificant size relative to the association probabilities, play an essential role in the dynamics of the IL-1β network. Without the dissociation probabilities, we have two possible final outcomes for the system, the signalling and nonsignalling ternary complexes T and NT; however, with the dissociation probabilities greater than zero, the probability of the Markov process arriving at the signalling ternary complex T is $\pi_4 \approx 1$.

6 Conclusion

A stochastic model of the IL-1β network has been presented. The behaviour of the Markov process has been described both computationally and analytically. IL-1β is a very active cytokine, requiring only tens of receptors to invoke a cellular response. Many inhibitory control mechanisms have evolved alongside the IL-1β network, possibly due to its potency [11]. One inhibitory control mechanism, the type-II nonsignalling receptor, has been modelled in the Markov chain analysed above.

It was found that this model has a unique stationary distribution in which the system occupies the ternary signalling complex with probability close to one. Given the relative size of the association and dissociation rates of the type-I and type-II receptors, this is perhaps unsurprising, since IL-1β is a fundamentally

important cytokine. This has implications for the biology of the network. The potency of IL-1β is such that, without some form of inhibition, it could potentially cause considerable damage.

It has been argued that the type-II receptor acts as a decoy, or *sink*, for IL-1β [8], by absorbing any of it that has not yet bound to type-I receptors. However, the Markov chain model has shown that the type-II receptor is not, in the long run, an effective competitor for IL-1β, or for receptor accessory protein. Rather, its presence acts more to *delay* the network going to the signalling ternary complex state.

The *modelled* network exhibited a stationary distribution, but could a *biological* system be said to have an equilibrium? This is a complex question, which requires more analysis than we can really provide here; however, Jit, Henderson, Stevens and Seymour have examined this question in relation to the cytokine TNF-α. They found that rheumatoid arthritis has an equilibrium which is sustained by a low, but persistent, level of TNF-α, whereas systemic inflammatory response syndrome (SIRS) is fundamentally a nonequilibrium condition [14].

The dissociation rates of the system are extremely small, relative to the association rates. When the dissociation rates were set to zero, an exploration of the Markov chain behaviour revealed that, without the possibility of dissociation, the system no longer has a unique stationary distribution. It appears that the dissociation rates, despite their apparent numerical insignificance, have a significant role to play in the dynamics of the IL-1β network.

Acknowledgements. Many thanks to are due to Thurston Park for his constant encouragement and inspiration.

References

1. Auron, P.E., et al.: Nucleotide sequence of human monocyte interleukin 1 precursor cDNA. PNAS 81, 7907–7911 (1984)
2. Auron, P.E.: The interleukin 1 receptor: ligand interactions and signal transduction. Cytokine Growth Factor 9, 221–237 (1998)
3. Bandman, O., et al.: Complexity of inflammatory responses in endothelial cells and vascular smooth muscle cells determined by microarray analysis. Ann. N.Y. Acad. Sci. 975, 77–90 (2002)
4. Bensimon, C., et al.: A monoclonal antibody recognizing 68- to 75-kilodalton protein(s) associated with the human IL-1 receptor. J. Immunol. 142, 2290–2298 (1989)
5. Bensimon, C., et al.: Two distinct affinity binding sites for IL-1 on human cell lines. J. Immunol. 143, 1168–1174 (1989)
6. Horuk, R., et al.: A biochemical and kinetic analysis of the interleukin-1 receptor. Evidence for differences in molecular properties of IL-1 receptors. J. Biol. Chem. 262, 16275–16278 (1987)
7. Boraschi, D., et al.: Mapping of receptor binding sites on IL-1 beta by reconstruction of IL-1ra-like domains. J. Immunol. 155, 4719–4725 (1995)
8. Colotta, F., et al.: The type II decoy receptor: a novel regulatory pathway for interleukin 1. Immunol. Today 15, 562–566 (1994)
9. Dinarello, C.A.: Biology of interleukin-1. J. FASEB, 108–115 (1988)

10. Dinarello, C.A.: Interleukin-1, interleukin-1 receptors and interleukin-1 receptor antagonist. Int. Rev. Immunol. 16, 457–499 (1998)
11. Eisenberg, S.P., et al.: Interleukin 1 receptor antagonist is a member of the interleukin 1 gene family: evolution of a cytokine control mechanism. Proc. Natl. Acad. Sci. U.S.A. 88, 5232–5236 (1991)
12. Greenfeder, S.A., et al.: Molecular cloning and characterization of a second subunit of the interleukin 1 receptor complex. J. Biol. Chem. 270, 13757–13765 (1995)
13. Grimmett, G., Stirzaker, D.: Probability and random processes. Oxford University Press, Oxford (2001)
14. Jit, M., Henderson, B., Stevens, M., Seymour, R.: TNF-α neutralization in cytokine-driven diseases: a mathematical model to account for therapeutic success in rheumatoid arthritis but therapeutic failure in systemic inflammatory response syndrome. Rheumatology 44, 323–331 (2005)
15. Jurič, D.M., Čarman-Kržan, M.: Interleukin-1β, but not IL-1α, mediates nerve growth factor secretion from rat astrocytes via type I IL-1 receptor. Int. J. Dev. Neurosci. 19, 675–683 (2001)
16. Matsushima, K., et al.: Properties of a specific IL-1 receptor on human EBV-transformed B lymphocytes: identity of the receptor for IL-1a and IL-1β. J. Immunol. 136, 4496–4502 (1986)
17. Park, T.: Non-Euclidean graphs. Publicationes mathematicae (Debrecen) 6, 290–297 (1959)
18. Poole, S., Gaines Das, R.E.: The international standards for interleukin 1α and interleukin 1β. Evaluation in an international collaborative study. J. Immunol. Methods 142, 1–13 (1991)
19. Schotanus, K., et al.: Domains of rat interleukin 1 beta involved in type I receptor binding. Endocrinology 36, 332–339 (1995)
20. Sims, J.E., Dower, S.K.: Interleukin-1 receptors. Eur. Cytokine Netw. 5, 539–546 (1994)
21. Smith, D., et al.: The Soluble Form of IL-1 Receptor Accessory Protein Enhances the Ability of Soluble Type II IL-1 Receptor to Inhibit IL-1 Action. Immunity 18, 87–96 (2003)
22. Wesche, H., et al.: The interleukin-1 receptor accessory protein (IL-1RAcP) is essential for IL-1-induced activation of interleukin-1 receptor-associated kinase (IRAK) and stress-activated protein kinases (SAP kinases). J. Biol. Chem. 272, 7727–7731 (1997)

Modelling the Tunability of Early T Cell Signalling Events

Nick D.L. Owens[1], Jon Timmis[1,2], Andrew Greensted[1], and Andy Tyrrell[1]

[1] Department of Electronics, University of York, UK
{ndlo100,jt512,ajg112,amt}@ohm.york.ac.uk
[2] Department of Computer Science, University of York, UK

Abstract. The Tunable Activation Threshold hypothesis of T Cells is investigated through computational modelling of T cell signalling pathways. Modelling techniques involving the π-calculus and the PRISM model checker are presented, and are applied to produce a stochastic model of T cell signalling. Initial results which demonstrate tuning of T cells are presented.

1 Introduction

The T lymphocyte has a major role in the response of the adaptive immune system. Through the T cell receptor (TCR) the T Cell responds to populations of antigenic peptide presented by the major histocompatibility complex molecule (pMHC) on nucleated cells [12]. The ability of a T cell to correctly discriminate and respond is remarkable given that the TCR is essentially randomly generated through sommatic mutations [12], and that foreign pMHCs will often only be 0.01% to 0.1% of the total expressed by a cell [4], the other 99.9% – 99.99% being self.

A number of discrepancies between observable immunology and classical clonal selection theory suggest that the classical theory does not give a complete picture. For example, there is clear evidence of self-reactive T Cells in the periphery and that T cells require interaction with self for survival in the periphery [4]. In light of this, a number of theories have arisen to explain the successful operation of T Cells.

One theory of particular interest is the tunable activation threshold hypothesis (TAT) presented by Grossman and his colleagues [1], [2]. The theory proposes that lymphocytes adapt their activation thresholds based upon recent interactions with their environment. As such, the T cells tune to local interactions and react to a change in the environment rather than any one specific interaction. Such a mechanism would allow auto-reactive T Cells to exist in the periphery with high activation thresholds.

Altan-Bonnet and Germain (ABG) [5] model a particular pathway which has been shown to be involved in T Cell sensitivity [14], and has been implicated for tuning [4], [2]. The ABG model provides a starting point here, we investigate the pathway directly for tunability properties. The ABG model is re-implemented in a stochastic formulation using the stochastic π-calculus, simulated using the

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 12–23, 2008.

Stochastic Pi Machine (SPiM) [11] and converted for analysis in the PRISM probabilistic model checker [19].

This work also contributes to the design of new artificial immune systems [6], we intend to use to the model to build new immune inspired algorithms. This should clarify our choice of modelling methods: the model is in a computational language for a more ameanable transition toward algorithms. As no further steps are taken in this direction in this work we give it no further mention.

Our paper is structured as follows: section 2 biolgical background; section 3 contains a description of modelling methods; section 4 gives a modelling patterns that may be used to convert reaction based description of a biological model to a π-calculus model; section 5 outlines the model; finally section 6 provides some prelimary results.

2 Biological Background and Model

Peptide MHC TCR interactions can be classified by the response they illicit in a cell. We take the definitions of [4]:

- *Agonist.* Will induce all possible activation signals within a cell.
- *Partial agonist.* Will induce a subset of all possible activation signals within a cell.
- *Antagonist.* Will actively inhibit activation signals within the cell.
- *Null.* Will not have any affect, activatory or inhibitory.

A range of signal strengths exists for each of the first three classes, it is not the case that the signals induced by a weak agonist are necessarily stronger than those induced by a partial agonist. Self-peptides fall into either partial agonist and antagonist classes [4].

2.1 Signalling Components

Phosphorylation is a primary signalling mechanism in biological pathways, it involves the addition of a phosphate group to a molecule which changes its conformation and so its ability to bind to other molecules.

We now give an overview of the signalling components pertenant to this paper, for a full description see [12].

- T Cell receptor ζ-Chains are internal components of the TCR, typically there are two such chains. Each ζ-chain contains 3 Immunoreceptor tyrosine-based activation motifs (ITAMs). Each ITAM may be twice phosphorylated.
- Leukocyte-specific protein tyrosine kinase (Lck), may be soluble in the cytosol or associated with TCR co-receptor CD4/8. Lck phosphorylates ITAMS, SHP-1.
- CD4/8 TCR Co-receptor. Populations of TCR co-receptors are expressed on the surface of the T Cell. The co-receptor binds to TCR-pMHC complexes with a stabilising effect. The internally the co-receptor is associated with Lck which may phosphorylate internal TCR components upon co-receptor binding to TCR-pMHC complex.

- Cytosolic soluble ζ-chain associated protein kinase 70 (ZAP70) binds to a twice phosphorylated ITAM, protecting the phosphorylation. Processivity occurs with ZAP70 binding [4], there is a particular order in which it may bind to phosphorylated ITAMs. Fully ZAP70 laden ITAMs instigate further signalling pathways.
- SH2 domain containing tyrosine phosphatase (SHP-1) binds to internal TCR complex. It is phosphorylated by Lck to become pSHP-1 and dissociates. Soluble pSHP-1 may rebind TCR internal complex and upon further Lck phosphorylation will dephosphorylate non-ZAP70 protected ITAMs.
- Extracellular signal-regulated kinase (ERK) is part of mitogen-activated protein kinase (MAPK) cascade and involved in T Cell effector signalling. It provides an indicator of the activation of the cell. A twice phosphorlated form of ERK may bind Lck in TCR complex modifying the SH2 domain [14] protecting the TCR internal chains from dephosphorylation by SHP-1.
- Mitogen-activated protein kinase (MAPK) cascade is a commonly found biological component [8]. In T cells the RAF-MEK-ERK MAPK cascade plays a role in T Cell activation [4]. The instigation of the MAPK cascade results in the twice phosphorylation of ERK.

There are three concepts central to the T Cell signalling described in this paper: kinetic proofreading, noise reducing negative feedback, and amplifying positive feedback, which are now discussed.

2.2 Kinetic Proofreading

Kinetic proofreading arose to describe the remarkable accuracy of DNA replication and protein synthesis [22]. McKeithan [7] applied kinetic proofreading to T cell signalling, and it is now a widely accepted model to account for ligand discrimination [14]. There are three key concepts to Kinetic proofreading. First, upon binding of pMHC to TCR there are a number of energy consuming signalling steps (they take physical amounts of time to overcome) instigated internally in the T Cell. All these steps must occur before T Cell activation signalling can start. Second, upon dissociation of pMHC from TCR these steps are rapidly reversed. Third, the greater the specificity between TCR and pMHC the longer the bind. The length of bind is regarded as one of the best measures of TCR-pMHC bind quality [13]. The kinetic proofreading steps measure the length of the bind and so the quality of the bind.

The T Cell signalling described in this paper contains two proofreading mechanisms, the phosphorylation of TCR internal chains and the association the TCR co-receptor.

- **Phosphorylation of ITAMs and ZAP70.** Activation signalling requires a fully ZAP70 laden ζ-chain, the bind between TCR-pMHC must be long enough to allow full phosphorylation of the ITAMs and the processivity of ZAP70 binding mentioned in section 2.1
- **Association of Co-receptor.** This is not a necessary proofreading mechanism, as it is possible for an activation signal to propagate from a TCR

without a co-receptor bound. However, the co-receptor will stabilise the TCR-pMHC complex [15] and co-receptor associated Lck will phosphorylate TCR internal chains far more efficiently than soluable Lck [14]. Thus a TCR-pMHC complex that exists for long enough to allow co-receptor association will be far more likely to overcome ITAM phosphorylation proofreading and so is more likely successfully produce an activation signal.

2.3 Noise Reducing Negative Feedback

Kinetic proofreading alone is not sufficient to explain antagonism [13]. A negative feedback effect, investigated experimentally in [14] may augment proofreading to compensate for its shortfallings. After TCR engagement SHP-1 may bind to TCR internal complex and be phosphorylated by Lck. Phosphorylated SHP-1 (pSHP-1) dissociates and may re-associate to a potentially different TCR, allowing the pSHP-1 signal to spread. This reassociated pSHP-1 may be further phosphorylated, which activates pSHP-1 causing it to desphosphorylate any non ZAP70-protected ITAMs. This creates a negative feedback which dampens any activatory signal.

2.4 Amplifying Positive Feedback

The combination of proofreading and negative feedback alone would mean that the level of negative feedback should increase as the quality of the TCR-pMHC bind increases. This is not the case: there is a point as ligand quality increases where the pSHP-1 negative signal disappears [14]. An explanation exists through the protecting effect of double phosphorylated ERK (ppERK). A TCR complex which sucessfully overcomes proofreading and negative feedback will instigate the MAPK cascade [1]. The result of the MAPK cascade is ppERK which protects the TCR complex from the dephosphorylating effect of pSHP-1 and so breaking the negative feedback loop. In a similar manner to pSHP-1, ppErk will spread allowing TCR to protect the signalling of other surrounding TCRs.

These three mechanisms provide a mapping between TCR-pMHC bind time and cell signalling such that there is a natural discrimination between pMHC ligands, in order of decreasing bind quality:

1. Agonist – Binds TCR long enough such that there is a high probability of induce MAPK cascade in the face of a pSHP-1 dampening. It will receive protection from ppERK.
2. Antagonist – Binds long enough to induce high levels of pSHP-1 but not to induce MAPK cascade.
3. Partial Agonist/Endogenous Ligand – Does not bind long enough to induce high levels of pSHP-1, but may induce some partial phosphorylation of TCR internal chains. This is where self should lie [13].

[1] The MAPK cascade is a robust signalling pathway found in many cells throughout biology, it has been shown to have an all-or-none amplification effect [8].

The spreading of the pSHP-1 and ppERK signals is of interest here. An antagonist that induces high levels of pSHP-1 will dampen the activation signal propagating from the surrounding TCR. An agonist may protect the surrounding TCR allowing the agonist to synergise with non-agonist ligands and spread the protection signal. This mechanism provides an explanation of how a signal agonist in a sea of non-agonist ligands is able to induce the T Cell into activation.

The interplay between SHP-1 and ERK as a candidate for tunability has been suggested in [4], [2], [5], [13]. In this context the level of pSHP-1 indicates the size of negative feedback and defines a functional threshold that must be overcome for signalling. It is these ideas that are investigated in the remains of the paper.

There is related work in T Cell, particularly TAT modelling, the pathways described above are modelled in [5], [15], examples of TAT models include [1], [3].

3 Modelling Methods

There is perhaps growing similarity between computational and biological systems in terms of concurrency, distributively, connectivity and particularly the discrete rather than continuous nature of both systems. As a consequence concurrent computational analysis techniques have been applied to biology by a number of authors, e.g. [11], [18], [19]. This is the approach here, a description of the biological model is developed in the stochastic π-calculus which is then analysed through simulation and conversion into a model checker.

A simulation tool provides a single instantiation of the system, the stochastic fluctuations inherent in the simulation method will mean that the system will behave differently on subsequent runs. For example a system with an unknown bistablilty will arrive in either of its stable states at the end a simulation, at least two runs are necessary to simply determine that the system is bistable. A probabilistic model checker overcomes the need for multiple simulations by allowing analysis of the probability distributions of the system. Here we perform such analysis by employing the PRISM model checker [19].

3.1 The Stochastic π-Calculus

The π-calculus developed by Milner and colleagues [9] is a process algebra used to describe concurrent computational systems with the property of mobility. The Stochastic π-calculus first described by Priami [10] applies stochastic extensions to the π-calculus replacing non-determinism with race conditions defined by exponential distributions. As such the stochastic π-calculus allows quantitative analysis of π-calculus systems through a mapping to an underlying continuous time markov chain [10].

The stochastic π-calculus was originally applied to biology in [18]. There are now a range of tools for analysis of biological models described with stochastic π. Particularly there are two simulators BioSpi [18] and SPiM [11], both employ the Gillespie algorithm to guarantee correct chemical kinetics. SPiM is used here to simulate our model.

In a process algebra, such as the π-calculus, processes communicate concurrent over shared channels. Applying this paradigm to biology a process P is a molecule, a channel c describes the existence of a reaction between two processes. For the π-calculus the reaction rules specify which reactions may proceed, the stochastic race conditions in conjunction with mass-action kinetics, see section 3.2, dictate the timings of the reactions.

A variant of the stochastic π-calculus presented in [11] is given, this is the stochastic π-calculus used at the core of SPiM.

Definition 1. *The syntax π-calculus definition*[2]*.*

$$P ::= \mathbf{0} \mid \pi.P \mid P + Q \mid (P \mid Q) \mid \nu x P \mid \star P \tag{1}$$

Left to right: null | action prefix | choice | parallel[3] *| restriction | replication.*

Action Prefixes:

$$?x_r(\tilde{y})\ \textit{input} \quad | \quad !x_r(\tilde{y})\ \textit{output} \quad | \quad \tau_r\ \textit{silent/delay} \tag{2}$$

r denotes the rate of the prefix. $\tilde{y}$ denotes a tuple may be sent of received along a channel.

Definition 2. *Structural Congruence on π-calculus. Alpha-conversion (change of bound names); identity $P \mid \mathbf{0} \equiv P$; commutativity of parallel and choice $P \mid Q \equiv Q \mid P$, $P+Q \equiv Q+P$; associativity of parallel $P \mid (Q \mid R) \equiv (P \mid Q)|R$; Identity of restriction $\nu x\mathbf{0} \equiv \mathbf{0}$; commutativity of restriction $\nu x \nu y P \equiv \nu y \nu x P$; scope extrusion $\nu x(P \mid Q) \equiv P \mid \nu x Q$* **if** *$x \notin \mathrm{fn}(P)$, the free names in P; $\star P \equiv P|\star P$ defintion of replication.*

Definition 3. *Reduction rules:*

$$\tau_r.P + M \xrightarrow{r} P \tag{3}$$

$$!x_r(\tilde{n}).P + M \mid ?x_r(\tilde{m}).Q + N \xrightarrow{r} P \mid Q_{\{\tilde{n}/\tilde{m}\}} \tag{4}$$

$$P \xrightarrow{r} P' \Rightarrow \nu x P \xrightarrow{r} \nu x P' \tag{5}$$

$$P \xrightarrow{r} P' \Rightarrow P \mid Q \xrightarrow{r} P' \mid Q \tag{6}$$

$$P \equiv Q, P' \equiv Q', P \xrightarrow{r} P' \Rightarrow Q \xrightarrow{r} Q' \tag{7}$$

3.2 Gillespie Algorithm

The Gillespie algorithm [16] is rigorously derived from a stochastic formulation of chemical kinetics, it performs a Monte-Carlo simulation of a chemical system. Given a chemical system of molecules, a set of reactions, and a state (i.e. the

[2] This differs slightly from the syntax given in [11], this is for the sake of brevity, all essential components are given here.

[3] The parenthesis are not syntactically necessary for parallel composition, it is just to aid clarity of the use of | within the BNF definition.

populations sizes of reactants), the Gillespie algorithm will determine which reaction occurs next and when it occurs. The law of mass-action states that a reaction proceeds proportional to a deterministic rate d and the number of reactants. For a molecule type X, the number of molecules in a reaction system is denoted |X|. There are three reactions that are of concern here:

Type	Reaction Equation	Rate	Example
1st Order	$\mathrm{X} \xrightarrow{d} \mathrm{Y}$	$d\lvert\mathrm{X}\rvert$	Degradation
2nd Order	$\mathrm{X}_1 + \mathrm{X}_2 \xrightarrow{d} \mathrm{X}_3$	$d\lvert\mathrm{X}_1\rvert\lvert\mathrm{X}_2\rvert$	Complexation
2nd Order Symmetric	$\mathrm{X}_1 + \mathrm{X}_1 \xrightarrow{d} \mathrm{X}_3$	$\frac{d}{2}\left(\lvert\mathrm{X}_1\rvert - 1\right)\lvert\mathrm{X}_1\rvert$	Homodimerisation

The rate of second order reactions are defined by the number of possible pairs of reactants. For SPiM's implementation of the Gillespie algorithm [11] the rates of first order and second order reactions are implemented directly, however for symmetric reactions the modeller must compensate and divide rates by two.

It is necessary to convert a deterministic rate d of second order reaction to a stochastic rate c for use with the Gillespie algorithm. The rates must be scaled by the volume, V, of the reaction system and should one want to simulate exact numbers of molecules then the rate is also be scaled by the Avogadro Number $N_A = 6.022 \times 10^{23}$.

$$c = \frac{r}{N_A V} \tag{8}$$

For computational complexity purposes it may be necessary to simulate a fraction $\epsilon \in [0, 1]$ of a volume V, for a concentration C of a molecule the rate now becomes $c = r/N_A V\epsilon$ and the number of molecules in the simulation $N = CN_A V\epsilon$.

3.3 PRISM Probabilistic Model Checker

Probabilistic Model Checking is a formal analysis technique used to assert the quantitative correctness of models of systems, it requires a formal description of the system and a specification of a system in a temporal logic. In PRISM [19] this is done with a description of the system as a continuous time markov chain (CTMC) and an extension of temporal logic CSL. States of the markov chain are augmented with rewards and the CSL can be used to calculate expected rewards both transiently and in the steady state. The only CSL queries described here are of the form that are used later in the model: $\mathcal{R}_{=?}[I = t]$ – Describes what is the expected reward at time instant t.

4 Modelling Patterns

Modelling abstractions for first order reactions, second order complexations and enzymatic reactions are described, such that if these are applied to a system of reactions one should achieve the correct stochastic π-calculus formalism of the reaction system.

4.1 First and Second Order Reactions: Degradation, Complexation, Enzymatic Reactions

In general a first and second order reactions can be described by the following programs:

Reaction	Program	Instantiation
$\mathrm{X} \xrightarrow{r} \mathrm{Y}$	$X = \tau_r.Y$	X
$\mathrm{X}_1 + \mathrm{X}_2 \xrightarrow{r} \mathrm{X}_3$	$X_1 = ?c_r.X_3,\ X_2 = !c_r$	$\nu c_r(X_1 \mid X_2)$
$\mathrm{X}_1 + \mathrm{X}_1 \xrightarrow{r} \mathrm{X}_3$	$X_1 = ?c_r.X_3 + !c_r$	$\nu c_r X_1$

Complexation has been modelled in π-calculus by the communication over a shared channel [11], for example in the reaction $\mathrm{X} + \mathrm{Y} \xrightarrow{r} \mathrm{XY}$:

$$X = \nu p(!c_r(p).X_b) \qquad Y = ?c_r(p).Y_b \tag{9}$$

X_b and Y_b are the bound states and X and Y respectively. If X and Y are initiated in parallel they will react to privately share channel p which X_b and Y_b may communicate on.

Enzymatic reactions occur between an enzyme E acting on a substrate S to form a product P, they compromise two stages:

$$\mathrm{E} + \mathrm{S} \underset{k_d}{\overset{k_a}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{k_c} \mathrm{E} + \mathrm{P} \tag{10}$$

The enzyme will bind to the substrate at association rate k_a, they may dissociate at rate k_d or the enzyme will convert the substrate into the product at rate k_c. This is often modelled [11] as:

$$E = \nu d_{k_d} \nu c_{k_c}\ !a_{k_a}(d_{k_d}, c_{k_c}).(?d_{k_d}.E + ?c_{k_c}.E) \tag{11}$$

$$S = ?a_{k_a}(d_{k_d}, c_{k_c}).(!d_{k_d}.S + !c_{k_c}.P) \tag{12}$$

Where E and S share private channels with differing rates, the race condition in the choice dictates whether the enzyme is successful in producing the product. However a different formulation of enzymatic reactions is more appropriate here:

$$E \ = \ ?a_{k_a}.(\tau_{k_d}.(E|S) + \tau_{k_c}.(E|P)) \qquad S \ = \ !a_{k_a} \tag{13}$$

This formulation sacrifices the ability to exchange information on a private channel between enzyme and substrate, but it reduces the number of required processes from 4 to 3 and the number of required channels from 3 to 1. The two formulations are behaviourly equivalent by bisimulation, this can be straightforwardly proved using the approaches given in [20]. Under simulation and analysis the formulations will behave identically in terms of visible populations of enzyme, substrate and product. The reason for the change in formulation is twofold, first, SPiM contains optimisations based on assumptions that simulations will contain

large numbers of identical of process [11]. For 100 simulation runs of the MAPK cascade, model outlined in section 5, the optimisations translate to the formulation of equation 13 running in ~ 3 minutes and the formulation of equation 11 running in ~ 6 minutes. Second, the formulation is more similar to the process calculi used [17] (allowing automata translation into PRISM) and in [20] (allowing reduction of state space for analysis). Such automated translation and state space reduction are not used in this work but will be a major part of future analysis work.

5 The Model

The model is in essence a stochastic formulation of the ABG model [5], with minor changes. The CD8 co-receptor is only allowed to bind to the TCR-pMHC complex and pMHC. In the ABG model the following complexes are allowed CD8-TCR, CD8-pMHC, CD8-TCR-pMHC. The change clarifies the proofreading behaviour of CD8 and is in line with the model of Wylie [15].

The model proceeds as the biology is described in section 2. There is a further simplification in line with ABG, only a single ζ-chain with three ITAMs are included. Each ITAM may be twice phosphorylated allowing and requiring binding of 3 ZAP70 molecules to signal the MAPK cascade. The reactions used can be found online at [21], which for the majority are identical to the reactions found in the ABG model.

The π-calculus model is generated mechanistically from the reactions in [21] using the patterns defined in section 3. The volume of simulation is scaled by a factor 100, this is performed without any qualitative loss to the results, the ABG model contains 3×10^4 TCRs, 300 are simulated here. The π-calculus model, runnable in SPiM can be found online [21].

Since PRISM performs analysis on the entire state space of a system it is not computationally tractable to perform model checking on a population of TCRs and pMHCs. The model checker is used to investigate the signalling behaviour of a single TCR-pMHC complex by volume restriction to one TCR, even so it is still necessary to restrict the model even further to reduce the state space. The inclusion of the MAPK cascade give rise to a system of 224613312 states and 3703035840 transitions. If it is removed and ppERK protection replaced by a boolean parameter we achieve a far more tractable system of 2050 states and 11312 transitions. The number of pSHP-1 molecules are also supplied as a parameter, the PRISM model can be found [21].

6 Model Simulation and Analysis

To demonstrate tunability given the hypotheses presented in this paper it first must be shown that TCR-pMHC interactions result in a higher concentration of cytosolic pSHP1. Second it must be shown that this higher concentration of pSHP1 hinders the ability of the TCR to signal. We must allow ensure that the

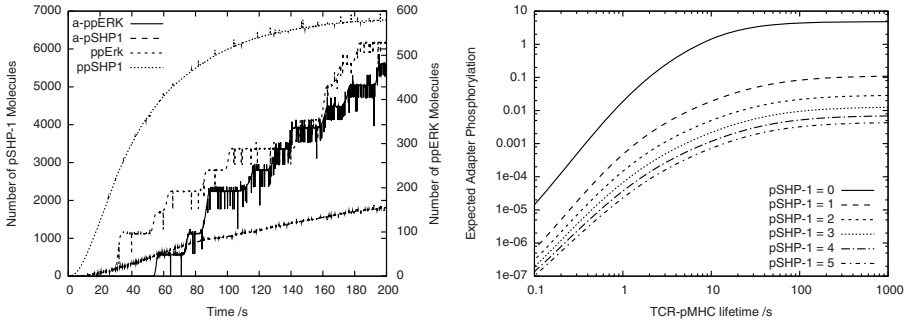


Fig. 1. Right: Number of pSHP-1 and ERK molecules for 10 agonist ligands ($k_d = 18s$) alone: a-ppERK and a-pSHP-1, and then with the addition of 1000 non-agonist/antagonist ligands ($k_d = 3s$): ppERK and pSHP-1. Right: expected levels of Adapter protein phosphorylation at $t = 200s$ for vary ligand quality and for differing levels of pSHP-1, the 0 level represents TCR protection, no inhibition from pSHP-1.

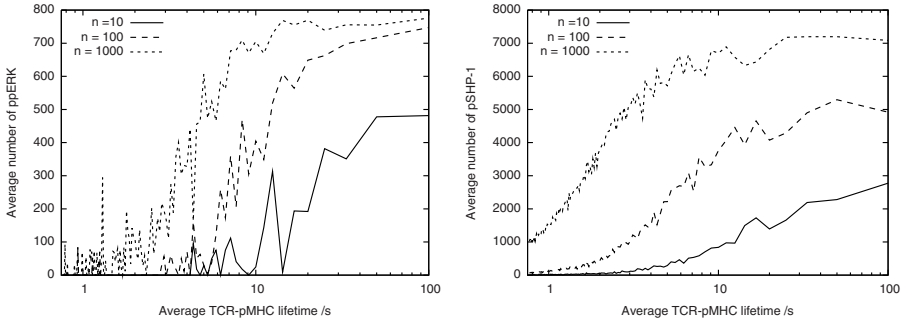


Fig. 2. Average number of ppERK (left) and pSHP1 (right) molecules across a 200 second simulation with $n = 10, 100, 1000$ pMHC complexes

model of the T cell must display the correct speed, selectivity and sensitivity as outlined by [13].

To demonstrate that higher levels of pSHP-1 hinder the ability of the TCR to signal we employed the PRISM model. Figure 1 (right) shows the expected adapter phosphorylation against ligand quality at $t = 200s$ after binding, greater levels of pSHP-1 massively decrease the expected adapter phosphorylation. It is important to note that the PRISM model is scaled to a single TCR so a change of 0 to 4 pSHP-1 molecules in the vaccinity of that TCR represents a massive change in pSHP-1 concentration in the entire cell. Second, the PRISM model does not contain the MAPK cascade and so lacks the ability of the TCR to protect itself, should protection by ppERK occur the expected phosphorylation of the Adapter will immediately jump to the pSHP-1= 0 line.

The remaining tests are performed using the simulations of the π-calculus model. Figure 2 shows the average number of ppERK and pSHP1 molecules

over a 200 second simulation of $n = 10, 100, 1000$ pMHC molecules across a range of binding affinities. The jaggedness is facet of taking an average of just 10 simulation runs. The results demonstrate that, as expected, at higher binding affinities a larger amount of pSHP-1 is produced. However the pSHP-1 levels flatten off as more ppERK is successfully produced (a higher probability of a pSHP-1 protected TCR.

A further interesting result is that of synergy, figure 1 (left) shows that the presenting non-agonist ligands with agonist ligands increases both pSHP-1 and ppERK levels. The large increase of both suggests that further parameter analysis will reveal antagonism.

7 Conclusions

We have outlined the biological components of a signalling system which exhibits tuning properties. We present computational modelling methods and patterns that are generally applicable to any reaction based biological system. We apply these to the biology presented and gain stochastic π-calculus and PRISM models. We show some preliminary results, demonstrating tuning and synergy. However, during runs of the model antagonism appears elusive (data not shown), this may be a facet of the stochastic formulation identified by [15], and a further biological mechanism may be required. We shall continue with further model analysis and investigations into new immune inspired algorithms.

Acknowledgments

This work is sponsored by EPSRC Grant Number: EP/E005187/1.

References

1. Grossman, Z., Paul, W.E.: Adaptive Cellular Interactions in the Immune System: The Tunable activation threshold and significance of subthreshold. PNAS 89, 10365–10369 (1992)
2. Grossman, Z., Paul, W.E.: Autoreactivity, dynamic tuning and selectivity. Current Opinion in Immunology 13, 687–698 (2001)
3. Carneiro, J., Paixoa, T., Milutinovicb, D., Sousaa, J., Leona, K., Gardnera, R., Faroa, J.: Immunological self-tolerance: Lessons from mathematical modeling. Journal of Computational and Applied Mathematics 184, 77–100 (2005)
4. Germain, R.N., Stefanov, I.: The dynamics of T cell receptor signaling: complex orchestration and the key roles of tempo and cooperation. Annu. Rev. Immunol. 17, 467–522 (1999)
5. Altan-Bonnet, G., Germain, R.N.: Modeling T cell antigen discrimination based on feedback control of digital ERK responses. PLoS Biol. 3, e356 (2005)
6. Timmis, J., Andrews, P., Owens, N., Clark, E.: An Interdisciplinary Perspective on Artificial Immune Systems. Evolutionary Intelligence 1, 5–26 (2008)
7. McKeithan, T.W.: Kinetic proofreading in T-cell receptor signal transduction. PNAS 92, 5042–5046 (1995)

8. Huang, C.Y., Ferrell, J.E.: Ultrasensitivity in the mitogen-activated protein kinase cascade. PNAS 93, 10078–10083 (1996)
9. Milner, R.: Communicating and Mobile Systems: the π-Calculus. Cambridge University Press, Cambridge (1999)
10. Priami, C.: Stochastic π-Calculus. The Computer Journal 38, 578–589 (1995)
11. Phillips, A., Cardelli, L.: Efficient Correct Simulation of Biological Processes in Stocastic Pi-calculus. In: Calder, M., Gilmore, S. (eds.) CMSB 2007. LNCS (LNBI), vol. 4695, pp. 184–199. Springer, Heidelberg (2007)
12. Janeway, C., Travers, P., Walport, M., Shlomchik, M.: Immunobiology: The Immune System is Health and Disease Garland Science (2004)
13. Feinerman, O., Germain, R.N., Altan-Bonnet, G.: Quantitative challenges in understanding ligand discrimination by alphabeta T cells. Mol. Immunol. 45, 619–631 (2008)
14. Štefanová, I., Hemmer, B., Vergelli, M., Martin, R., Biddison, W.E., Germain, R.N.: TCR ligand discrimination is enforced by competing ERK positive and SHP-1 negative feedback pathways. Nat. Immunol. 4, 248–254 (2003)
15. Wylie, D.C., Das, J., Chakraborty, A.K.: Sensitivity of T cells to antigen and antagonism emerges from differential regulation of the same molecular signaling module. PNAS 104, 5533–5538 (2007)
16. Gillespie, D.T.: Exact Stochastic Simulation of Coupled Chemical Reactions. The Journal of Physical Chemistry 81, 2340–2361 (1977)
17. Norman, G., Palamidessi, C., Parker, D., Wu, P.: Model checking the probabilistic π-calculus. In: Proc. QEST 2007, pp. 169–178. IEEE Computer Society, Los Alamitos (2007)
18. Priami, C., Regev, A., Shapiro, E.: Application of a stocastic name-passing calculus to representation for biological processes in the stochastic π-calculus. Information Processing Letters 80, 25–31 (2001)
19. Heath, J., Kwiatkowska, M., Norman, G., Parker, D., Tymchyshyn, O.: Probabilistic model checking of complex biological pathways. Theoretical Computer Science 319, 239–257 (2008)
20. Hermanns, H.: Interactive Markov Chains. LNCS, vol. 2428. Springer, Heidelberg (2002)
21. Model description: Reactions, SPiM and PRISM code, http://www-users.york.ac.uk/~ndlo100/icaris2008/
22. Hopfield, J.J.: Kinetic Proofreading: A New Mechanism for Reducing Errors in Biosynthetic Processes Requiring High Specificity. PNAS 71(10), 4135–4139 (1974)

Immune Responses: A Stochastic Model

Anastasio Salazar-Bañuelos[1,2]

[1] Hotchkiss Brain Institute
[2] Department of Surgery, Division of Transplantation
University of Calgary, AB, Canada
salazara@ucalgary.ca

Abstract. Immune phenomena are explained from the reductionist view of the immune system as a collection of cells, molecules, and their interactions. Although this approach has produced abundant valuable information, it has added increased complexity. Artificial Immune Systems (AIS) have relied on this theoretical framework to emulate the desired characteristics of immunity. However, the complexity of the theoretical base has lead to an impasse in AIS research, suggesting that a new theoretical framework is needed. A theoretical model is presented here that explains immune responses as a "swarm function". The model proposes a system based on two stochastic networks: a central recursive network, wherein the proportion of agents is determined and maintained, and a peripheral network, wherein the random interactions of these agents determine if an inflammatory response will emerge from the system.

1 Introduction

There is a recognition that a different theoretical framework is needed in the field of Artificial Immune Systems (AIS) [1,2]. It has been proposed that the failure to generate practical results in AIS is a result of the high degree of complexity of the human immune system, and the use of less complex primitive immune systems has been advocated [3]. While not denying that simpler approaches may lead to more practical solutions, I will argue here that the problem is of a more conceptual nature.

1.1 The Problem

AIS construction has been inspired by predominant paradigms in immunology, consequently it has focused on Self/Non-Self discrimination, clonal selection, danger signals, etc. However, these theories are insufficient to explain some biological phenomena [4,5]. As well, the research focus on cellular and molecular mechanisms has led to constant discoveries of an increasingly complex set of agents and interactions, and yet there will be always some unknown interaction and element not considered in our knowledge of the system. For example, the relatively recent discovery of T regulatory lymphocytes and their importance in the concept of dominant tolerance [6,7] has changed our previous understanding of autoimmunity and the concept of Self [8]. Adding to this, the cells and

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 24–35, 2008.

molecules involved in immune responses also participate in a variety of biological phenomena not always related to immunity and have been subjected to a different evolutionary pressures. As a result, we find that immune responses to viruses, bacteria, parasites, auto-antigens, allo-antigens, xeno-antigens, cancer antigens, or simple inflammation as result of a minor trauma, cannot be explained by the same mechanism, even when these phenomena share many cellular and molecular components and pathways.

To understand the immune phenomena at cellular and molecular level, we will need an immunological Laplace's Demon, that is, the ability to know the complete set of members and interactions of the entire immune system at any given time. Similarly, to create an AIS upon this knowledge, we will need unlimited computer power to include all the interactions taking place in real time in this biological jungle. Only then can we predict why and when an immune response occurs.

Immune Responses and the Immune System. We think of immune responses as being the function of the immune system. However, as difficult as it may seem, the immune system is a poorly defined and poorly delimited system, which sometimes even includes the skin and the colonic flora. Over the 20th century, the immune system has come to include all the cells and molecules associated with destroying pathogens, Non-Self antigens, and harmful agents. As consequence, we tend to consider pathogen-driven responses, autoimmunity, transplant rejection, cancer responses, allergy, etc., as intrinsic properties of these cells and molecules. However, immune responses are macroscopic phenomena, not merely cellular or molecular events. In other words, autoimmunity is not the presence of auto-reactive clones or auto-antibodies, transplant rejection is not the presence of anti-HLA antibodies or allo-reactive clones, etc. An acute episode of autoimmunity, the rejection of an organ, the destruction of a tumour, or the inflammation of a traumatized tissue are inflammatory processes directed towards an specific antigen or group of antigens. These inflammatory processes are a colony function, wherein all components participate, including the antigen and the microenvironment where the reactions take place. Whereas the study of individual molecules and cells can help explain how the inflammatory process propagates, it does not explain why the inflammatory process occurred in the first place or in one particular location and time rather than another, such as an acute episode of autoimmunity or an acute rejection episode in a transplanted organ, which are discrete events emerging without an apparent direct cause. In contrast, inflammation can be absent even though reactive clones and auto-antibodies are present [9], indicating that the presence of these cells and molecules, although necessary, is insufficient to explain the emergence of the immune response.

A holistic approach to the immune system dates back to Elie Metchnikoff at the end of the 19th century and the origins of immunology as a scientific discipline. However the reductionist approach centred on the study of cells, molecules, and their mechanisms currently prevails [10], with the notorious exception of Jerne's Network Theory [11] and further contributions [12,13,14,15,16] but these have so far failed to produce convincing data to constitute a practical alternative

[17]. Despite this, there is increased interest in finding a new theoretical framework at the system level that will explain immune responses [18].

Defining the System. Many immunological systems are well understood at the molecular and cellular levels, such as the generation of antibody repertoires [19], clonal selection [20], and the HLA system [21,22]. The problem is that these mechanisms fail to answer a critical question in immune-driven phenomena: why does it occur in the first place? There are associations, such as the presence of anti-HLA antibodies with transplant rejection [23] and the presence of auto-antibodies with autoimmunity; however, presence of these immune agents only represents an increased probability that the phenomena will occur, whereas the phenomena itself is unpredictable. For example, we do not know why or how an acute attack of autoimmunity is triggered or if or when an acute rejection episode will take place, but we do know that if the patient has been sensitized to the antigen or has detectable auto-antibodies or autoreactive clones, then it is more likely that a response will occur. Interestingly, therapeutic interventions with immunosuppressants, which modify the cellular and molecular components of the inflammatory response [24], diminish the frequency of acute attacks in autoimmunity and transplant rejection, but do not abrogate the phenomena. Here, we consider that the system that needs to be defined is one that can delimit the occurrence of an immune response as a phenomenal, rather than as a statistical entity.

2 The Model

Instead of trying to explain the functions of a conceptually ill-defined immune system, I will focus on a model that tries to explain the phenomena of inflammation as a common pathway of immune responses.

2.1 Immune Phenomena Emerge from Stochastic Events

Any immune-mediated process, irrespective of its evolutionary history or present risk factors, ends in an identifiable phenomena that we name inflammation. It is a self-propagating phenomenon taking place at local peripheral (tissue) level that causes injury to cells, molecules, or other materials bearing a Self (native, original) or Non-Self (post-natal, de novo) antigen. The system responsible for this phenomenon consists of cells and molecules from the lymphatic system, which function as independent agents that interact in a random fashion between each other, the local environment, and the antigens (Fig. 1), creating a self-reproducing complex adaptive system. These random interactions consist of pro-inflammatory and inhibitory events, which neutralize each other in normal (healthy) conditions, keeping the system in a non-inflammatory mode. In contrast, inflammatory responses emerge from this system of stochastic events as an escalation of positive feedback loops of non-random events, such as the liberation of mediators, homing of cells, activation of enzymatic systems, proliferation of specific clones etc. directed towards an antigen.

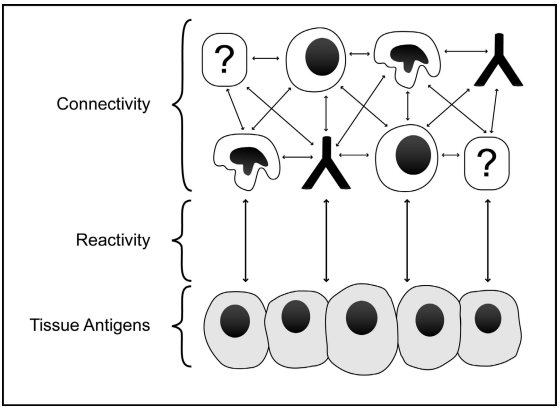


Fig. 1. A highly interconnected network of agents of the lymphatic system in their local microenvironment and their interaction with antigens. The type and intensity of these interactions constitutes reactivity toward the antigen. Question marks indicate unknown (undiscovered) agents or interactions.

2.2 Distance from the Phase Transition Defines the Probability of an Inflammatory Reaction

We can assume then, that the event that determines the inflammatory process is a change in the behaviour of the system from random self-neutralizing interactions to a nonrandom, escalated, and self-maintained cascade of pro-inflammatory events. As a consequence, the delimitation between normal auto-reactivity versus autoimmunity, tolerance versus rejection, etc., is the phase transition that separates the two modes of the system. It follows then, that a single pro-inflammatory event can trigger an inflammatory response if it makes the system reach the threshold separating these two modes. Therefore, the probability of an inflammatory process depends on the distance between the status of the system at any given time and a critical point at which the threshold is reached, rather than on the pro-inflammatory event or agent itself (Fig. 2).

2.3 Agents Can be Simplified as Pro-inflammatory, Anti-inflammatory, or Neutral

We can simplify the system by defining the agents by their contribution to making the system either closer to or farther from the threshold (pro-inflammatory, anti-inflammatory, or neutral) and ignoring the diverse and complex processes that created these agents. These assumptions can be abstracted to apply to unknown agents or interactions, as well as to properties of a cell or molecule, which can be inhibitory, stimulatory, or neutral according to particular circumstances. By reducing the system to these three (in practice two) interactions, we can escape the need to incorporate special functions and properties of each agent

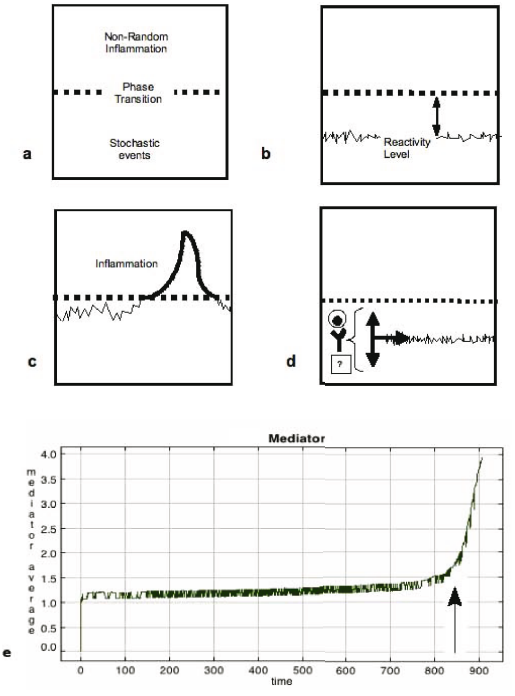


Fig. 2. Phases in the development of an inflammatory response. (a) At the peripheral level, the system is composed of two phases, the first characterized by random self-neutralizing interactions between pro- and anti-inflammatory events and the second by an escalation of pro-inflammatory events separated by a phase transition. (b) The distance between the reactivity level and a threshold defining the phase transition is what determines the probability for an inflammatory response, not the antigen or independent agent by itself. (c) Inflammatory responses emerge from this system once the threshold is reached. (d) The only influence in determining the distance between the reactivity level and the threshold that an agent can have is by increasing, decreasing, or being indifferent (arrows). The question mark indicates unknown influences that can be assumed to fall in one of these three categories. (e) Computer simulation showing the emergence of a new dynamic in the system after reaching a critical value. The graph represents the average mediator produced by 2 independent agents interacting randomly in a two-dimensional space. One agent increases, while the other decreases, the concentration of mediator in its local environment. Both populations increase their numbers until they reach a critical level, after which the production of the mediator escalates, indicating the emergence of new dynamics in the system, shown by the arrow.

(antigen-presenting cells, blocking antibodies, HLA molecules, etc.), eluding the problem of the incompleteness of empirical data.

To summarize thus far, we have a complex adaptive system acting in peripheral tissue, where inflammatory reactions emerge once a threshold is reached by the contribution of competing pro-inflammatory and anti-inflammatory (suppressive)

mechanisms. This threshold divides the system in two phases: a phase characterized by random interactions wherein microscopic events are mutually neutralized and an inflammatory phase wherein the system behaves as a positive feedback loop of proinflammatory, non-random events.

2.4 Recursion Can Explain Robustness and Diversity

Although this theoretical construction can explain how immune responses emerge from a stochastic network of interactions between cells and molecules, it does not explain why immune responses are antigen-specific, why they show broad variability in their intensity (not only among individuals but within the same individual in different circumstances or at different stages of development), or why they tend to resist modification once they are established, a property known as robustness. For this construction to explain the specificity, diversity, and robustness of immune responses, several conditions must apply. First, there must be as many levels of reactivity (specificity) as antigens exposed to the system. Second, the distance between the level of reactivity of the system to the threshold must be set at different levels among individuals (inter-individual variation) and be susceptible to variation within the same individual (intra-individual variation). Third, the level of reactivity must show a strong tendency to return at a fixed point in the status of the system (robustness).

How can the level of reactivity for each antigen be robust, specific, and show variation among individuals, yet permit modification in order to explain tolerance to previously rejected antigens and autoimmunity to originally accepted antigens? Vaz and Varela [12] suggested that the answer may be by recursion; their example is reproduced by computer simulation in Fig. 3. A ball is picked randomly from a box containing one white and one black ball. The ball is placed back into the box along with another ball of the same colour. By repeating this action until there are a large number of balls in the box, we can observe that the proportion of black and white balls will show an initial fluctuation in the range between 0 and 1, with a further stabilization relative to the number of balls in the box. The more balls, the less fluctuations, following a power law where the stability of the proportions is directly related to the number of events. If we consider one colour as pro-inflammatory and the other as anti-inflammatory (suppressor), and by their interaction they neutralize each other, we can simulate the predominance influence as well as the intensity of that influence or reactivity level (Fig. 3). We will end with a situation where sometimes one colour will predominate over the other, pro-inflammatory influences on the reactivity level will predominate over suppressive ones or vice versa, and this predominance will also vary in intensity, given the different distances between the reactivity level and the phase transition or threshold.

Instead of balls, let us now take items of different shapes (representing different antigens), each shape having a black and a white version (representing pro- and anti-inflammatory influences). In this scenario, a specific proportion for each shape and colour will result, and increasing the number of iterations will lead to stability of the reactive level. If we substitute these items for lymphocyte clones

specific for an antigen (different shape) with pro-inflammatory and suppressive (black and white) versions, then we will get a situation where the proportion of clones will be stable for each antigen.

The lymphatic system consists of billions of cells. The daily turnover of cells in the bone marrow is in the order of 500 billion; of these, 20 billion are lymphocytes directly related to immune responses, with an approximately average life span of 4.4 years [25]. Because enormous numbers of cells are constantly regenerated, it offers an ideal situation for a recursive process to take place in the maintenance of cellular clones at specific and stable proportions. This can explain the establishment of reactivity levels for several different antigens and may well explain why, despite the fact that the cells of the lymphatic system are replaced constantly, the immune responses tend to be stable and reliable. However, despite the fact that immune responses are robust, they can be modified by biological manipulation, as is the case in vaccination-induced immunity or in radiation-induced autoimmunity [26]. In the case of vaccination-induced immunity, a new antigen is introduced where a level of reactivity for this particular antigen has not been set. According to the present model, a recursive process will start to take place, and the type of response will be determined in the early stages of the recursive process. The way the antigen is presented, rather than the antigen itself, is what will determine further responses, exemplified by some well known phenomena such as high and low zone tolerance, induction of tolerance or sensitization depending the administration route, and the induction of immunity or disease depending on the type of adjuvant given. In the case of radiation-induced autoimmunity, the previous reactivity level towards the tolerated original (Self) antigen is reset by depleting the lymphocyte populations. This would be equivalent to randomly remove large numbers of balls from the box in the previous example, placing the system (depending on the number of balls left) in a stage where more fluctuations can be expected. As a consequence, it is possible to achieve different long-term reactivity levels, such as the induction of a response towards a previously tolerated antigen or the abrogation of a previously determined response (Fig. 3).

3 The Model and Immunological Processes

3.1 Clonal Selection

Although the clonal selection theory explains how clonal selection takes place, as somatic hypermutation explains the generation of antibody diversity, it does not explain the immune response in autoimmunity, cancer, transplant rejection, etc. The contribution of clonal selection to the immune response is to influence the level of reactivity for each antigen; in other words, it biases the distance between the status of the system and the threshold by eliminating auto-reactive clones in the early ontogeny of the lymphatic system. Viewed in this way, the present model explains the apparent contradiction between the existence of auto-reactive clones or auto-antibodies and the absence of an inflammatory process. If clonal selection works by its relative contribution to the level of reactivity, it can be

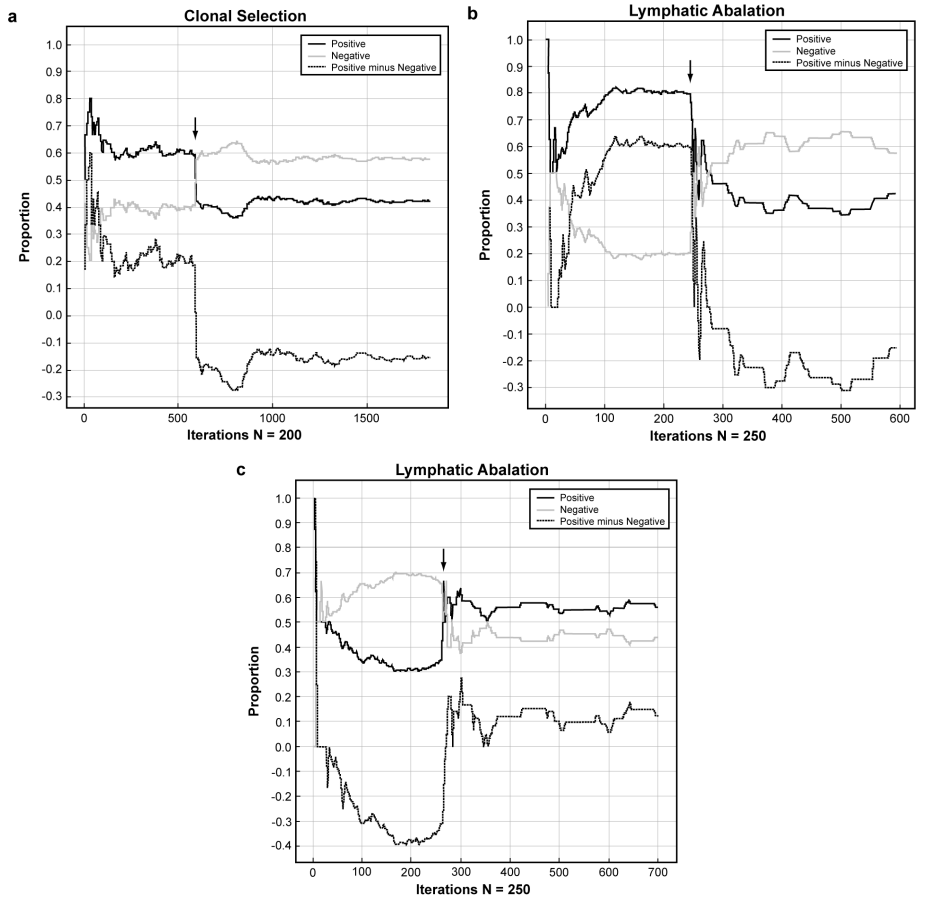


Fig. 3. Computer simulation of 3 recursion experiments. The solid black line represents the proportion of positive (pro-inflammatory) agents, the solid gray line represents the proportion of negative (anti-inflammatory) agents, and the dotted line represents the difference between the two, indicating the intensity of the predominant influence or reactivity level. (a) Elimination of pro-inflammatory agents at an early stage (arrow) simulates negative selection of auto-reactive clones. As a consequence, the level of reactivity is biased to a predominance of negative agents, setting the level distant from the threshold. This simulates how clonal selection biases the recursive process towards suppression of auto-reactive clones to define Self. (b, c) Elimination of all agents after stability had been reached (arrow), except one pro-inflammatory and one anti-inflammatory agent, which recapitulate the initial step of the recursive process. This creates the conditions for setting reactivity at a different level. Change can occur in either direction, increasing or decreasing the distance to the threshold. This models possible outcomes after an adult individual is subjected to profound lymphatic ablation. (b) The reactivity level is set farther from the threshold, indicating possible remission of autoimmune disease. (c) The reactivity level is set closer to the threshold, indicating possible induction of autoimmune disease.

(as it is) incomplete. In other words, the complete elimination of auto-reactive clones is not required to avoid autoimmunity, and the same situation stands for the generation of auto-antibodies.

3.2 Determination of Self

In the pre-natal and early post-natal period of development, the thymus generates the conditions for the deletion of auto-reactive clones [27] and the survival of tissue-specific T regulatory cells [7], biasing the development of lymphatic system towards predominance of suppressive agents directed to the antigens presented in the thymus (Fig 3a). Because of the isolation of the thymus to external antigens [28], by default, these thymic antigens will be the inborn set of the individual. This will set the level of reactivity at a distance from the threshold compatible with what we call recognition of Self. In other words, Self will be defined by setting the reactivity level to a distance that prevents reaching the threshold spontaneously. The distance to the reactivity level will be maintained by recursion, particularly in the bone marrow, accounting for the consistency displayed by immune responses. We call this consistency immunological memory.

3.3 Autoimmunity

Autoimmunity occurs more frequently in the post-thymus period of development, after a period of tolerance to the antigen towards which the autoimmunity develops. Interestingly, depletion of the lymphatic system is associated both with the generation [26] and the cure [29,30] of autoimmune diseases, as well as the development of allograft donor-specific unresponsiveness in mixed leukocyte reaction tests after total lymphoid irradiation [31]. Depletion of lymphatic cells is the equivalent of randomly removing a large number of balls from the box from the previously explained recursive experiment to a degree that causes the stability level to be set at different proportions of pro- and anti-inflammatory events (Fig 3). Because this change can produce either a decrease (Fig. 3b) or increase (Fig. 3c) in the proportions of pro-inflammatory versus anti-inflammatory agents (lymphocytes), this model can explain the two opposite, seemingly paradoxical, phenomena of remission or induction of autoimmune diseases by radiation or by bone marrow transplants [32,33].

Self will become Non-Self producing autoimmunity when the reactivity level is set closer to the threshold by modification of the systems wherein the recursive mechanisms are taking place (principally in the bone marrow) by influences such as radiation, viral infections, chemicals, etc. Conversely, Non-Self may become Self when the reactivity level is set to predominantly suppressive influences (sub-threshold level). For example, the initial rejection and followed by hyporesponsiveness observed clinically after the introduction of a new antigen, particularly one not subjected to evolutionary pressure as in the case of allograft transplants, can be explained by the initial fluctuations in the recursive process. It is noteworthy that a high degree of immunosuppression, which usually includes a lymphocyte-ablating agent at the time of the implantation of the organ (induction), is required to engraft the transplanted organ.

4 Abstracting the System

The model presented here is composed of two complex adaptive systems, the first being a network of cells (autonomous agents) interacting and regenerating constantly (autopoiesis), where recursion is taking place for each antigen presented to the system and where new antigens in context are presented or represented (internal image). This system will produce pro- and anti-inflammatory agents which will migrate to a second network consisting of the independent agents, the antigens and the microenvironment. This system will receive other local influences that will determine fluctuations of the reactivity level at the local peripheral tissue, and inflammatory responses will emerge once the level of reactivity reaches the threshold. Although the reactivity level can be influenced by external factors including trauma, infection, etc., the reactivity level will tend to return to the level determined by the recursive system, which will work as an attractor for the second peripheral system (tissue).

5 Implications

Although the present model simplifies the complexity of cellular and molecular interactions, it does not contradict current theories at cellular and molecular levels. Rather, they are incorporated as part of the process for setting the threshold level for each antigen. The model also offers an explanation for the elusive problem of demarcation in immune responses, proposing a mechanism that can explain the divergent point between auto-reactivity and autoimmunity, tolerance and rejection, response and no response. The model also explains paradoxes in real-life phenomena that cannot be satisfactory explained by classical theories, and at the same time is coherent with the diversity shown by immune responses both in physiologic as well as pathologic conditions. Here, rather than being the cause for the immune response, the belonging of an antigen to a Self or Non-Self category is determined by the immune response after the fact. For AIS, this model is not based in naive metaphors and is not finalistic. More importantly, the model can be explored from the network theory point of view, such as a Hopfield Model.

Acknowledgements

I thank Luis Benítez-Bribiesca and Samuel Weibe for critical discussion of the model and Ann Kyle for editorial review.

References

1. Hart, E., Timmis, J.: Application areas of AIS: The past, the present and the future. Appl. Soft Comput. 8, 191–201 (2008)
2. Timmis, J.: Artificial immune systems—today and tomorrow. Natural Computing: an international journal 6, 1–18 (2007)

3. Twycross, J., Aickelin, U.: Biological inspiration for artificial immune systems. In: Sixth International Conference on Artificial Immune Systems, Santos, SP, Brazil (2007)
4. Coutinho, A.: The Le Douarin phenomenon: a shift in the paradigm of developmental selftolerance. Int. J. Dev. Biol. 49, 131–136 (2005)
5. Coutinho, A.: Immunology at the crossroads. As decades of research have resulted in few clinical applications, it is time to think about new research strategies to understand the workings of the immune system. EMBO Rep. 3, 1008–1011 (2002)
6. Sakaguchi, S., Sakaguchi, N., Asano, M., Itoh, M., Toda, M.: Immunologic self-tolerance maintained by activated T cells expressing IL-2 receptor alpha-chains (CD25). Breakdown of a single mechanism of self-tolerance causes various autoimmune diseases. J. Immunol. 155, 1151–1164 (1995)
7. Sakaguchi, S.: Naturally arising Foxp3-expressing CD25+CD4+ regulatory T cells in immunological tolerance to self and non-self. Nat. Immunol. 6, 345–352 (2005)
8. Schwartz, R.H.: Natural regulatory T cells and self-tolerance. Nat. Immunol. 6, 327–330 (2005)
9. Lacroix-Desmazes, S., Kaveri, S.V., Mouthon, L., Ayouba, A., Malanchere, E., Coutinho, A., Kazatchkine, M.D.: Self-reactive antibodies (natural autoantibodies) in healthy individuals. J. Immunol. Methods 216, 117–137 (1998)
10. Tauber, A.I.: The Immune Self: Theory or Metaphor? Cambridge University Press, Cambridge (1994)
11. Jerne, N.K.: Towards a network theory of the immune system. Ann. Immunol (Paris) 125C, 373 (1974)
12. Vaz, N.M., Varela, F.J.: Self and non-sense: an organism-centered approach to immunology. Med. Hypotheses 4, 231–267 (1978)
13. Stewart, J., Varela, F.J., Coutinho, A.: The relationship between connectivity and tolerance as revealed by computer simulation of the immune network: some lessons for an understanding of autoimmunity. J. Autoimmun. 2(suppl.), 15–23 (1989)
14. Stewart, J., Varela, F.J.: Morphogenesis in shape-space. Elementary metadynamics in a model of the immune network. J. Theor. Biol. 153, 477–498 (1991)
15. Varela, F.J., Coutinho, A.: Second generation immune networks. Immunol. Today 12, 159–166 (1991)
16. Stewart, J., Coutinho, A.: The affirmation of self: a new perspective on the immune system. Artif. Life 10, 261–276 (2004)
17. Carneiro, J.: Towards a comprehensive view of the immune system Ph.D. Thesis, University of Porto (1997)
18. Orosz, C.G., Forrest, S., Hoffmeyr, S., Cohen, I.R., Segel, L.A.: How complexity helps to shape alloimmunity. Graft 4, 365–382 (2001)
19. Tonegawa, S.: Somatic generation of antibody diversity. Nature 302, 575–581 (1983)
20. Burnet, F.M.: A modification of Jerne's theory of antibody production using the concept of clonal section. Aust. J. Sci. 20, 67–69 (1957)
21. Klein, J., Sato, A.: The HLA system. First of two parts. N. Engl. J. Med. 343, 702–709 (2000)
22. Klein, J., Sato, A.: The HLA system. Second of two parts. N. Engl. J. Med. 343, 782–786 (2000)
23. McKenna, R.M., Takemoto, S.K., Terasaki, P.I.: Anti-HLA antibodies after solid organ transplantation. Transplantation 69, 319–326 (2000)
24. Taylor, A.L., Watson, C.J., Bradley, J.A.: Immunosuppressive agents in solid organ transplantation: Mechanisms of action and therapeutic efficacy. Crit. Rev. Oncol. Hematol. 56, 23–46 (2005)

25. Fliedner, T.M., Graessle, D., Paulsen, C., Reimers, K.: Structure and function of bone marrow hemopoiesis: mechanisms of response to ionizing radiation exposure. Cancer Biother. Radiopharm. 17, 405–426 (2002)
26. Sakaguchi, N., Miyai, K., Sakaguchi, S.: Ionizing radiation and autoimmunity. Induction of autoimmune disease in mice by high dose fractionated total lymphoid irradiation and its prevention by inoculating normal T cells. J. Immunol. 152, 2586–2595 (1994)
27. von Boehmer, H., Teh, H.S., Kisielow, P.: The thymus selects the useful, neglects the useless and destroys the harmful. Immunol. Today 10, 57–61 (1989)
28. Castillo, A., Razquin, B., Villena, A.J., Zapata, A.G., Lopes-Fierro, P.: Thymic barriers to antigen entry during the post-hatching development of the thymus of rainbow trout, Oncorhynchus mykiss. Fish Shellfish Immunol. 8, 157–170 (1998)
29. Strober, S.: Total lymphoid irradiation in alloimmunity and autoimmunity. J. Pediatr. 111, 1051–1055 (1987)
30. Syed, M.I., Clark, L.J., Sturrock, R.D.: Unintended benefits of immunosupression on autoimmune disease due to chemoradiation therapy for head and neck cancer. Am. J. Otolaryngol. 29, 63–65 (2008)
31. Chow, D., Saper, V., Strober, S.: Renal transplant patients treated with total lymphoid irradiation show specific unresponsiveness to donor antigens the mixed leukocyte reaction (MLR). J. Immunol. 138, 3746–3750 (1987)
32. Sherer, Y., Shoenfeld, Y.: Autoimmune diseases and autoimmunity post-bone marrow transplantation. Bone Marrow Transplant 22, 873–881 (1998)
33. Mandalfino, P., Rice, G., Smith, A., Klein, J.L., Rystedt, L., Ebers, G.C.: Bone marrow transplantation in multiple sclerosis. J. Neurol. 247, 691–695 (2000)

Adaptive Spam Detection Inspired by a Cross-Regulation Model of Immune Dynamics: A Study of Concept Drift

Alaa Abi-Haidar[1] and Luis M. Rocha[2]

[1] Department of Informatics, Indiana University, Bloomington IN 47401, USA
aabihaid@indiana.edu
[2] Instituto Gulbenkian de Ciência, Oeiras, Portugal

Abstract. This paper proposes a novel solution to spam detection inspired by a model of the adaptive immune system known as the cross-regulation model. We report on the testing of a preliminary algorithm on six e-mail corpora. We also compare our results statically and dynamically with those obtained by the Naive Bayes classifier and another binary classification method we developed previously for biomedical text-mining applications. We show that the cross-regulation model is competitive against those and thus promising as a bio-inspired algorithm for spam detection in particular, and binary classification in general.

1 Introduction

Spam detection is a binary classification problem in which e-mail is classified as either ham (legitimate e-mail) or spam (illegitimate or fraudulent e-mail). Spam is very dynamic in terms of advertising new products and finding new ways to defeat anti-spam filters. The challenge in spam detection is to find the appropriate threshold between ham and spam leading to the smallest number of misclassifications, especially of legitimate e-mail (false negatives). To avoid confusions, ham and spam will be labeled as negatives and positives respectively.

The vertebrate adaptive immune system, which is one of the most complex and adaptive biological systems, learns to distinguish harmless from harmful substances (known as pathogens) such as viruses and bacteria that intrude the body. These pathogens often evolve new mechanisms to attack the body and its immune system, which in turn adapts and evolves to deal with changes in the repertoire of pathogen attacks. A weakly responsive immune system is vulnerable to attacks while an aggressive one can be harmful to the organism itself, causing autoimmunity. Given the conceptual similarity between the problems of spam and immunity, we investigate the applicability of the cross-regulation model of regulatory T-cell dynamics [5] to spam detection.

Spam detection has recently become an important problem with the ubiquity of e-mail and the rewards of no-cost advertisement that can reach the largest audience possible. Spam detection can target e-mail headers (e.g. sender, receiver, relay servers...) or content (e.g. subject, body). Machine learning techniques such

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 36–47, 2008.

as support vector machines [13], Naive Bayes classifiers [18, 15] and other classification rules such as Case-Based Reasoning [9] have been very successful in detecting spam in the past. However, they generally lack the ability to track *concept drift* since they rely on training on fixed corpora, features, and rules. Concept drift is the (gradual or sudden) change of thematic context (often reoccurring) over time such as new advertisement themes in spam and Bayesian poisening, a technique used by spammers to surpass bayesian based spam filters. Ideally, a system is capable of handling concept drift if it adapts quickly to the thematic change, distinguishing it from noise [19]. Research in spam detection is now focusing on detecting concept drifts in spam, with very promising results [7, 14]. Other areas of intense development in spam-detection are social-based spam detection models [4, 6] as well as algorithms based on Artificial Immune System (AIS) [17] (based on clonal selection) [3] (based on ABNET, an Anti-Body Network) [20] (based on incremental clustering Immune Networks). The AIS models are inspired by diverse responses and theories of the natural immune system [11] such as negative selection, clonal selection, danger theory and the immune network theory. Our bio-inspired spam detection algorithm is based instead on the cross-regulation model [5], which is a novel development in AIS approaches to spam detection. Since this dynamic model is quite compelling in the simplicity by which it achieves harmful/nonharmful[1] discrimination, we expect it to be useful in also in spam/ham e-mail classification. Moreover, its dynamic nature, in principle, makes it a good candidate algorithm to deal with concept drift in e-mail, which we start testing here.

Section 2 offers a short review of the cross-regulation model [5]. Section 3 presents the Cross-regulation Spam Algorithm—our bio-inspired cross-regulation algorithm—and its application to the spam classification problem. Section 4 discusses the experiments and implementation of the model vis a vis other binary classification models. Finally, in the last two sections, the discussion of the results and the conclusion follow.

2 The Cross-Regulation Model

The cross-regulation model, proposed by Carneiro et al. [5], aims to model the process of discriminating between harmless and harmful antigens[2]—typically harmless self/nonself and harmful nonself. The model consists of only three cell types: Effector T-Cells (E), Regulatory T-Cells (R) and Antigen Presenting Cells (A) whose populations interact dynamically, ultimately to detect harmful antigens. E and R are constantly produced, while A are capable of presenting a collection of antigens to the E and R. T-cell proliferation depends on the co-localization of E and R as they form conjugates (bind) with the antigens presented by A cells (this model assumes that A can form conjugates with a maximum of two E or R). The population dynamics rules of this model are

[1] Or less accurately but more commonly used, self/nonself discrimination.

[2] Antigens are foreign substances, usually proteins or protein fragments, that trigger immune responses.

defined by three differential equations, which can be, for every antigen being presented by an A, summarized by the following three laws of interaction:

1. If one or two E bind to antigen, they proliferate with a fixed rate.
2. If one or two R bind to the antigen, they remain in the population.
3. if an R binds together with an E to the same antigen, the R proliferates with a certain rate and the E remains in the population but does not proliferate.

Finally, the E and R die at a fixed death rate. Carneiro et al. [5] showed that the dynamics of this system leads to a bistable system of two possible stable population concentration attractors: (i) the co-existence of both E and R types identifying harmless self antigens, or (ii) the progressive disappearance of R, identifying harmful antigens.

3 The Cross-Regulation Spam Algorithm

In order to adopt the cross-regulation algorithm for spam detection, which we named the Immune Cross-Regulation Model (ICRM), one has to think of e-mails as analogous to the organic substances that upon entering the body are broken into constituent pieces by lysosome in A. In biology, these pieces are antigens (typically protein fragments) and in our bio-inspired algorithm they are words or features extracted from e-mail messages. Thus, in this model, antigens are words or potentially other features (e.g. bigrams, e-mail titles). For every antigen there exists a number of virtual E and R that interact with A, each associated with a specific e-mail message, and whose role is to present, in distinct slots, a sample of the features of the respective e-mail message. Therefore A, E and R have specific affinities $\rho \in \Sigma$, where $E_{\rho 1}$ and $R_{\rho 2}$ can bind to a slot of A, $A_{\rho 3}$, only if $\rho 1 = \rho 3$ and $\rho 2 = \rho 3$ respectively.

The general ICRM algorithm is designed to be first trained on N e-mails of "self" (a user's outbox) and harmless "nonself" (a user's inbox). However, in the results described here, it was not possible to directly obtain outbox data. We are working on collecting outbox data for future work. Similarly, the ICRM is also trained on "harmful nonself" (spam arriving to a given user). Training on or exposure to ham e-mails, in analogy with Carneiro's et al model [5], is supposed to lead to a "healthy" dynamics denoted by the co-existence of both E and R with more of the latter. In contrast, training on or exposure to spam e-mails is supposed to result in much higher numbers of E than R. When e-mail features occur for the first time, a fixed initial number of E and R, for every feature, are generated. These initial values of E and R are different in the training and testing stages; more weight to R for ham features, and more weight to E for spam features is given in the labeled training stage. While we specify different values for initializing the proportions of E and R associated with e-mail features, depending on whether the algorithm is in the training or the testing stage, the ICRM is based on the exact same algorithm in both stages. The ICRM algorithm begins when an e-mail is received and cycles through three phases for every received e-mail:

In the **pre-processing phase**, HTML tags are not stripped off and are treated as other words, as often done in spam-detection [15] . All words constituting the e-mail subject and body are lowercased and stemmed using Porter's algorithm after filtering out common English stop words and words of length less than 3 characters. A maximum of n processed unique features (words, in this case) are randomly sampled and presented by the virtual A which corresponds to the e-mail. These virtual antigen presenting cells have n_A binding slots (that E and R can bind to) per feature, i.e. $n \times n_A$ slots per e-mail message. The breaking up of the e-mail message into constituent portions (features) is inspired by the natural process in Biology, but is further enhanced in this model to select the first and last $\frac{n}{2}$ features in the e-mail. The assumption is that the most indicative information is in the beginning (e.g. subject) and the end of the e-mail (e.g. signature), especially concerning ham e-mails.

In the **interaction phase**, feature-specific R_g and E_f are allowed to bind to the corresponding antigens presented by A, which are arbitrarily (uniform random) located on its array of feature slots. Every adjacent pair of A slots is dealt with separately: the E_f for a given feature f proliferate only if they do not find themselves sharing the same adjacent pair of A binding slots with R_g, in which case only the R_g, associated with feature g, proliferate. The model assumes that novel ham features k tend to have their E_k suppressed by R_g of other pre-occurring ham features g because they tend to co-occur in the same message. As for the algorithm's parameters, let n_A be the number of A slots per feature. Let $(E_{0_{ham}}, R_{0_{ham}})$ and $(E_{0_{spam}}, R_{0_{spam}})$ be the initial values of E and R for features occurring for the first time in the training stage for ham and spam, respectively. For the testing stage, we have $(E_{0_{test}}, R_{0_{test}})$. Moreover, $E_{0_{ham}} << R_{0_{ham}}$, $E_{0_{spam}} > R_{0_{spam}}$ and $E_{0_{test}} > R_{0_{test}}$. In the ICRM implementation hereby presented, a major difference form Carneiro's et al model [5] was tried: the elimination of cell death. This is a rough attempt to provide the system with long term memory. Cell death can lead to the forgetfulness of spam or ham features if these features do not reoccur in a certain period of time as shown later section 4.

In the **decision phase**, the arriving e-mail is assessed based on the relative proportions of R and E for its n sampled features. Features with more R are assumed to correspond to ham while features with more E are more likely to correspond to spam. The proportions are then normalized to avoid decisions based on a few highly frequent features that could occur in both ham and spam classes. For every feature f, the feature score is computed as follows:

$$score_f = \frac{R_f - E_f}{\sqrt{R_f^2 + E_f^2}}, \tag{1}$$

indicating an unhealthy (spam) feature when $score_f \leq 0$ and a healthy (ham) one otherwise. $score_f$ varies between -1 and 1. For every e-mail message e, the e-mail immunity score is simply:

$$score_e = \sum_{\forall f \in e} score_f. \tag{2}$$

Note that a spam e-mail with no text such as as the cases of messages containing exclusively image and pdf files, which surpass many spam filters, would be classified as spam in this scheme—e-mail e is considered spam if $score_e = 0$. Similarly, e-mails with only a few features occurring for the first time, would share the same destiny, since the initial E is greater than R in the testing stage $E_{0_{test}} > R_{0_{test}}$ which would result in $score_e < 0$.

4 Results

E-mail Data. Given the assumption that personal e-mails (i.e. e-mails sent or received by one specific user) are more representative of a writing style, signature and themes, it would be preferable to test the ICRM on e-mails from a personal mailbox. Unfortunately, this is not offered by the most common spam corpus of *spamassasin*[3] and similarly for *ling-spam*[4]. In addition, the ICRM algorithm requires timestamped e-mails, since order of arrival affects final E/R populations. Timestamped data is also important for analyzing concept drifts over time, thus we cannot use the *PU1*[5] data described by Androutsopoulos et al. [2] . Delany's spam drift dataset[6], introduced by Delany et al. [8], meets the requirements in terms of timestamped and personal ham and spam however its features are hashed and therefore it is not easy to make tangible conclusions based on their semantics. The *enron-spam*[7] preprocessed data perfectly meets the requirements as it has six personal mailboxes made public after the enron scandal. The ham mailboxes belong to the employees *farmer-d, kaminski-v, kitchen-l, williams-w3, beck-s and lokay-m*. Combinations of five spam datasets were added to the ham data from *spamassassin* (s), *HoneyProject* (h), *Bruce Guenter* (b) and *Georgios Paliousras'* (g) spam corpora and then all six datasets were tokenized [15]. In practice, some spam e-mails are personalized, which unfortunately cannot be captured in this dataset since the spam data comes from different sources. Only the first 1500 e-mails of every enron are used in this experiment.

Evaluation. Two forms of evaluation were conducted: The first and more common in spam detection evaluation is the static or offline evaluation using K-fold cross validation [10] while the second is the dynamic or real-time evaluation using a sliding window that is particularly useful to study the model's capability of dealing with concept drifts in spam and/or ham over time.

[3] http://spamassassin.apache.org/publiccorpus/
[4] http://www.aueb.gr/users/ion/publications.html
[5] http://www.iit.demokritos.gr/skel/i-config/downloads/enron-spam/
[6] http://www.comp.dit.ie/sjdelany/Dataset.htm
[7] http://www.iit.demokritos.gr/ ionandr/publications/

In the **first evaluation**, for each of the six enron sets, we ran each algorithm 10 times. Each run consisted of 200 training (50% spam) and 200 testing or validation (50% spam) e-mails that follow in the timestamp order. From the 10 runs we computed variation statistics for the F-score[8], and Accuracy performance.

In the **second evaluation**, for each of the six enron sets, we trained each algorithm on the first 200 e-mails (50% spam) and then tested on a sliding window of 200 testing or validation (50% spam) e-mails that follow in the order of time the email was received. The sliding shift was 10 e-mails and the range was between e-mail 201 and e-mail 2800 resulting in 260 slides (from 1500 ham and 1500 spam only 100 ham and 100 spam are for training and the remaining 2800 are for validation). For every window we computed variation statistics of the percentage of FP (misclassified ham) and FN (misclassified spam) in addition to the F-score and Accuracy. We also performed a linear regression of the proportions of false positives and false negatives, %FP and %FN, using least squares and computed the slope coefficients, the coefficient of determination R^2 for each—for the purpose of evaluating the effect of concept drift if any.

ICRM Settings. In the e-mail pre-processing phase, we used $n = 50$, $n_A = 10$, $E_{0_{ham}} = 6$, $R_{0_{ham}} = 12$, $E_{0_{spam}} = 6$, $R_{0_{spam}} = 5$, $E_{0_{test}} = 6$ and $R_{0_{test}} = 5$. These initial E and R populations for features occurring for the first time are chosen based on the initial ratios chosen by Carneiro et al. [5] and were then empirically adjusted to achieve the best F-score and Accuracy results for the six enron datasets. Finally, the randomization seed was fixed in order to compare results to other algorithms and search for better parameters.

The ICRM was compared with two other algorithms that are explained in the following two subsections. The ICRM was also tested on shuffled (not in order of date received) validation sets to study the importance of e-mail reception order. The results are shown in table 1.

Naive Bayes (NB). We have chosen to compare our results with the multinomial Naive Bayes with boolean attributes [12] which has shown great success in previous research [15]. In order to fairly compare NB with ICRM, we selected the first and last unique $n = 50$ features. The Naive Bayes classifies an e-mail as spam in the testing phase if it satisfies the following condition:

$$\frac{p(c_{spam}).\prod_{f\in e-mail} p(f|c_{spam})}{p(c_{spam}).\sum_{c\in\{c_{spam},c_{ham}\}}\prod_{f\in e-mail} p(f|c)} > 0.5, \tag{3}$$

where f is the feature sampled from an e-mail, and $p(f|c_{spam})$ and $p(f|c_{ham})$ are the probabilities that this feature f is sampled from a spam and ham e-mail respectively, while c is the union of spam and ham e-mails. The results are shown in table 1 and plotted in figure 1.

[8] The F1-measure (or *F-Score*) is defined as $F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$, where $Precision = \frac{TP}{(TP+FP)}$ and $Recall = \frac{TP}{(TP+FN)}$ and $Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$ measures of the classification of each test set, where TP, TN, FP and FN denote true positives, true negatives, false positive and false negatives respectively [10].

Variable Trigonometric Threshold (VTT). We previously developed the VTT as a linear binary classification algorithm and implemented it as a protein-protein abstract classification tool[9] using bioliterature mining [1]. For more details please refer to [1]. The results are shown in table 1, plotted in figure 1.

Table 1. F-score and Accuracy mean +/- sdev of 10 runs for 50% spam enron data sets with the first three columns using ICRM (the first one applied on ordered e-mail, the second one on shuffled timestamps of testing data and the third on on ordered e-mail but with ICRM having cell death with death rate=0.02), the fourth one using Naive Bayes and the last one using VTT.

		ICRM			Other Algorithms	
Dataset		**Ordered**	**Shuffled**	**Cell Death**	**Naive Bayes**	**VTT**
Enron1	F-score	0.9 ± 0.03	0.9 ± 0.03	0.89 ± 0.03	0.89 ± 0.04	0.91 ± 0.04
	Accuracy	0.9 ± 0.03	0.9 ± 0.03	0.89 ± 0.04	0.87 ± 0.05	0.9 ± 0.04
Enron2	F-score	0.86 ± 0.06	0.85 ± 0.06	0.85 ± 0.05	0.92 ± 0.07	0.82 ± 0.23
	Accuracy	0.85 ± 0.06	0.83 ± 0.07	0.84 ± 0.05	0.93 ± 0.05	0.86 ± 0.13
Enron3	F-score	0.88 ± 0.04	0.88 ± 0.04	0.9 ± 0.03	0.93 ± 0.03	0.86 ± 0.08
	Accuracy	0.87 ± 0.05	0.87 ± 0.05	0.89 ± 0.04	0.92 ± 0.04	0.85 ± 0.07
Enron4	F-score	0.92 ± 0.05	0.92 ± 0.04	0.91 ± 0.06	0.92 ± 0.05	0.95 ± 0.03
	Accuracy	0.92 ± 0.05	0.92 ± 0.05	0.9 ± 0.07	0.91 ± 0.06	0.95 ± 0.03
Enron5	F-score	0.92 ± 0.03	0.87 ± 0.06	0.86 ± 0.04	0.94 ± 0.04	0.84 ± 0.13
	Accuracy	0.91 ± 0.03	0.87 ± 0.05	0.86 ± 0.05	0.95 ± 0.03	0.87 ± 0.09
Enron6	F-score	0.89 ± 0.04	0.9 ± 0.04	0.89 ± 0.03	0.91 ± 0.02	0.88 ± 0.05
	Accuracy	0.88 ± 0.05	0.89 ± 0.05	0.89 ± 0.04	0.9 ± 0.03	0.87 ± 0.07
Total	**F-score**	**0.9 ± 0.05**	**0.89 ± 0.05**	**0.88 ± 0.05**	**0.92 ± 0.04**	**0.88 ± 0.12**
	Accuracy	**0.89 ± 0.05**	**0.88 ± 0.06**	**0.88 ± 0.05**	**0.91 ± 0.05**	**0.88 ± 0.08**

Table 2. ICRM vs NB F-score and Accuracy for spam to ham ratio variations for all enrons

		50% spam	30% spam	70% spam
ICRM	F-score	0.9 ± 0.05	0.91 ± 0.03	0.79 ± 0.12
	Accuracy	0.89 ± 0.05	0.86 ± 0.05	0.83 ± 0.08
NB	F-score	0.92 ± 0.04	0.86 ± 0.07	0.79 ± 0.07
	Accuracy	0.91 ± 0.05	0.84 ± 0.07	0.74 ± 0.01

5 Discussion

Static Evaluation Results. As clearly shown in table 1 and figure 1, ICRM, NB and VTT are very competitive for most enron datasets, indeed the performance of ICRM is statistically indistinguishable from VTT (F-score and Accuracy p-values 0.15 and 0.63 for the paired t-test validating the null hypothesis of

[9] The Protein Interaction Abstract Relevance Evaluator (PIARE) tool is available at http://casci.informatics.indiana.edu/PIARE/

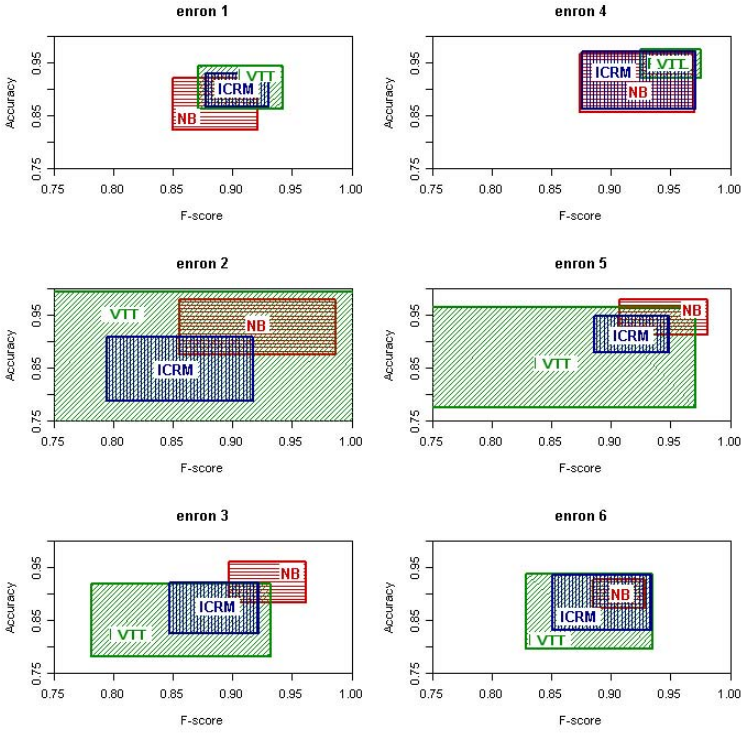


Fig. 1. F-score vs Accuracy mean and standard deviation plot comparison between ICRM (vertical blue), NB (horizontal red) and VTT (diagonal green) for each of the six enron datasets. A visualization of table 1

Table 3. ICRM vs NB F-score, accuracy, %FP and %FN slope coefficient ($\alpha_{\%FP}$ and $\alpha_{\%FN}$) and R^2, %FP and %FN for all enrons over time

Dataset		**F-score**	**Accuracy**	$\alpha_{\%FP}$,R^2	$\alpha_{\%FN}$, R^2	**%FP**	**%FN**
Enron1	ICRM	0.95 ± 0.01	0.95 ± 0.01	0.00,0.01	0.02,0.41	6.7 ± 1.5	4.11 ± 1.66
	NB	0.93 ± 0.01	0.93 ± 0.01	0.00,0.27	0.03,0.56	1.55 ± 0.53	12.99 ± 2.7
Enron2	ICRM	0.92 ± 0.01	0.92 ± 0.01	0.00,0.01	-0.01,0.11	6.48 ± 1.17	8.87 ± 1.89
	NB	0.95 ± 0.01	0.94 ± 0.01	0.01,0.10	0.00,0.01	9.57 ± 2.05	1.29 ± 0.48
Enron3	ICRM	0.93 ± 0.02	0.94 ± 0.02	0.03,0.95	0.01,0.20	4.7 ± 2.06	8.37 ± 1.77
	NB	0.92 ± 0.03	0.92 ± 0.02	0.00,0.43	0.05,0.52	0.51 ± 0.4	16.2 ± 5.2
Enron4	ICRM	0.92 ± 0.03	0.92 ± 0.03	0.04,0.52	0.03,0.37	6.99 ± 4.03	9.99 ± 2.92
	NB	0.92 ± 0.01	**0.93 ± 0.01**	0.00,0.56,	0.04,0.63	0.18 ± 0.27	15 ± 3.06
Enron5	ICRM	0.90 ± 0.02	0.90 ± 0.02	0.03,0.49	0.02,0.49	8.54 ± 2.58	12.08 ± 2.1
	NB	0.96 ± 0.03	0.96 ± 0.03	0.02,0.22	0.04,0.77	4.76 ± 3.44	3.06 ± 3.1
Enron6	ICRM	0.93 ± 0.01	**0.93 ± 0.02**	0.03,0.85	0.01,0.28	8.09 ± 2.23	5.33 ± 1.23
	NB	0.95 ± 0.01	0.95 ± 0.01	0.01,0.06	0.00,0.09	3.07 ± 2.17	6.89 ± 1.04

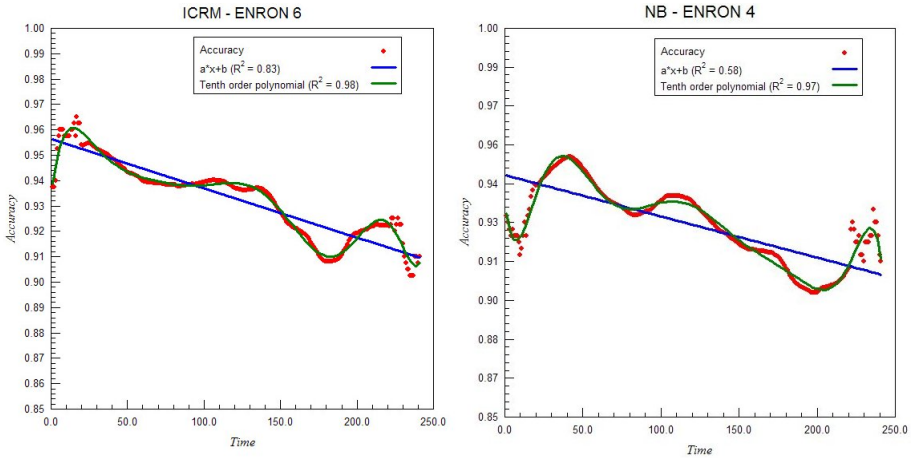


Fig. 2. ICRM Accuracy over time for enron6 and NB Accuracy over time for enron4, showing best linear and polynomial fits with R^2. The rest of the Accuracy and FN/FP plots are available as supplementary material.

variation equivalence), though its slightly lower performance against NB is statistically significant (F-score and Accuracy p-values 0.01 and 0.02 for the paired t-test, rejecting the null hypothesis of variation equivalence with 0.05 level of significance).

However, the ICRM can be more resilient to ham ratio variations[10] as shown in table 2. While the performance of both algorithms was comparable for 50% spam (though significantly better for NB), the performance of NB drops for 30% spam ratio (5% lower F-score than ICRM) and 70% spam ratio (9% less accurate than ICRM) while ICRM relatively maintains a good performance. The difference in performance is statistically significant, except for F-Score of the 70% spam experiment, as the p-values obtained for our performance measures clearly reject the null hypothesis of variation equivalence: F-Score and Accuracy p-values are 0 and 0.01 for 30% spam, and Accuracy p-value is 0.01 for 70% spam (p-value for F-Score is 0.5 for this case). While one could argue that NB's performance could well be increased, in the unbalanced spam/ham ratio experiments, by changing the right hand side of equation 3 to 0.3 or 0.7, this act would imply that, in real situations, one could know a priori the spam to ham ratio of a given user. The ICRM model, on the other hand, does not need to adjust any parameter for different spam ratios—it is automatically more reactive to whatever ratio it encounters. It has been shown that spam to ham ratios indeed vary widely [16, 8], hence we conclude that the ICRM's ability to better handle unknown spam to ham ratio variations is more preferable for dynamic data classification in general and spam detection in particular.

[10] The 30% and 70% spam results were balanced for the evaluation by randomly sampling from the 70% class, reducing it to 30%.

We have implemented ICRM with death rate[11] = 0.02. and without virtual cell death but the results showed negligible statistical differences (F-score and Accuracy p-values 0.02 and 0.04) although slightly in favor of no virtual cell death, as seen in table 1. The ICRM tested for spam variation and dynamic evaluation excluded cell death to speed up the algorithm, nonetheless, we are in the process of experimenting with heterogeneous death rates for the E, R cells of different features and more "interesting" features (e.g. e-mail title, from, to, and cc features). Since death rates affect the long-term memory of the system, this is something we intend to investigate more closely in future work.

In most Enron sets, shuffling the timestamps of received e-mails in the testing stages also only slightly reduced the ICRM's performance (F-score and Accuracy p-values 0.07 and 0.04 for paired t-test), therefore, unlike what was expected, the timestamps of e-mails seem to be largely irrelevant—which undermines some of the justification for a dynamic approach to spam detection based on the cross-regulation model. Nevertheless, we plan to study this further with additional data sets with much longer date ranges.

Dynamic Evaluation Results. The ICRM was also very competative with NB, have shown to be very competitive in the dynamic evaluation. The evidence is in the first two columns (F-score and Accuracy) of table 3 and in the supplementary material section[12].

Another notable feature of the ICRM is that it seems to balance %FN and %FP more efficiently over time. Conversely, NB tends to have high %FN and low %FP or vice versa. In order to quantify the balance between %FP and %FN, we compute their means and standard deviations for all enrons in the last two columns of table 3. While the largest mean in ICRM does not exceed 12.08% (enron 5), it does reach 12.99% (enron 1) 16.02% (enron 3) and 15% (enron 4) in NB for %FP. However, in spam detection in particular, more importance is given to %FP (ham misclassification) which favors NB over ICRM in most enron datasets. In future work, we will explore if enabling heterogeneous death rates for E and R cells can reduce %FP with the ICRM. On the other hand, the ICRM's more balanced %FN and %FP could be valuable for other binary classification problems where FP and FN are equally important—which is not the case in spam detection.

We also computed slope coefficients $\alpha_{\%FN}$, $\alpha_{\%FP}$ and their corresponding R^2 for the least square linear fit of %FN and %FP in order to study the behaviour of concept drift which would be manifested by high slopes—indicating decay in performance. However, the slopes are quite minimal as shown in the third and fourth columns of table 3. Indeed, the performance is essentially flat in time for both algorithms with slopes close to zero (see plots in supplemental materials). Therefore, there does not seem to be much concept drift in these datasets.

[11] Death rate = 0.02 resulted in the best performance for the death rate range $r \in [0.01, 0.1]$, where r is the probability that an R_f or E_f would die for a previously occurring feature f.

[12] All supplementary material is accessible at http://casci.informatics.indiana.edu/icaris08/

6 Conclusion

The observations made based on the artificial immune system can help us guide or further deepen our understanding of the natural immune system. For instance, ICRM's resilience to spam to ham ratio and its ability to balance between %FN and %FP show us how dynamic is our immune system and functional independently of the amount of pathogens attacking it. In addition, the three modifications made to the original model can be very insightful: The improvements made by training on both spam and ham (rather than only ham or self) reinforce the theories of both self and nonself antigen recognition by T-cells outside the thymus. The feature selection makes us wonder whether the actual T-cell to antigen binding is absolutely arbitrary. Finally, the elimination of cell death may reinforce the theories behind long lived cells as far as long term memory is concerned.

In this paper we have introduced a novel spam detection algorithm inspired by the cross-regulation model of the adaptive immune system. Our model has proved itself competitive with both spam binary classifiers and resilient to spam to ham ratio variations in particular. The overall results, even though not stellar, seem quite promising especially in the areas of spam to ham ratio variation and also of tracking concept drifts in spam detection. This original work should be regarded not only as a promising bio-inspired method that can be further developed and even integrated with other methods but also as a model that could help us better understand the behavior of the T-cell cross-regulation systems in particular, and the vertebrate natural immune system in general.

Acknowledgements. We thank Jorge Carneiro for his insights about applying ICRM on spam detection and his generous support and contribution for making this work possible. We also thank Florentino Fdez-Riverola for the very useful indications about spam datasets and work in the area of spam detection. We would also like to thank the FLAD Computational Biology Collaboratorium at the Gulbenkian Institute in Oeiras, Portugal, for hosting and providing facilities used to conduct part of this research.

Bibliography

[1] Abi-Haidar, A., Kaur, J., Maguitman, A., Radivojac, P., Retchsteiner, A., Verspoor, K., Wang, Z., Rocha, L.: Uncovering protein-protein interactions in abstracts and text using linear models and word proximity networks. Genome Biology (in press, 2008)

[2] Androutsopoulos, I., Koutsias, J., Chandrinos, K., Spyropoulos, C.: An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. ACM Press, New York (2000b)

[3] Bezerra, G., Barra, T.: An Immunological Filter for Spam. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 446–458. Springer, Heidelberg (2006)

[4] Boykin, P., Roychowdhury, V.: Leveraging social networks to fight spam. Computer 38(4), 61–68 (2005)

[5] Carneiro, J., Leon, K., Caramalho, Í., van den Dool, C., Gardner, R., Oliveira, V., Bergman, M., Sepúlveda, N., Paixão, T., Faro, J., et al.: When three is not a crowd: a Crossregulation Model of the dynamics and repertoire selection of regulatory CD4 T cells. Immunological Reviews 216(1), 48–68 (2007)
[6] Chirita, P., Diederich, J., Nejdl, W.: MailRank: using ranking for spam detection. In: Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 373–380 (2005)
[7] Delany, S.J., Cunningham, P., Smyth, B.: Ecue: A spam filter that uses machine leaming to track concept drift. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (eds.) ECAI 2006, 17th European Conference on Artificial Intelligence, PAIS 2006, Proceedings, pp. 627–631. IOS Press, Amsterdam (2006a)
[8] Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. Knowledge-Based Systems 18(4-5), 187–195 (2005)
[9] Fdez-Riverola, F., Iglesias, E., Díaz, F., Méndez, J., Corchado, J.: SpamHunting: An instance-based reasoning system for spam labelling and filtering. Decision Support Systems 43(3), 722–736 (2007)
[10] Feldman, R., Sanger, J.: The Text Mining Handbook: advanced approaches in analyzing unstructured data. Cambridge University Press, Cambridge (2006)
[11] Hofmeyr, S.: An Interpretative Introduction to the Immune System. Design Principles for the Immune System and Other Distributed Autonomous Systems (2001)
[12] Jensen, F., Jensen, F., Jensen, F.: Introduction to Bayesian Networks. Springer, New York (1996)
[13] Kolcz, A., Alspector, J.: SVM-based filtering of e-mail spam with content-specific misclassification costs. In: Proceedings of the TextDM, pp. 1–14 (2001)
[14] Méndez, J., Fdez-Riverola, F., Iglesias, E., Díaz, F., Corchado, J.: Tracking Concept Drift at Feature Selection Stage in SpamHunting: an Anti-Spam Instance-Based Reasoning System. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 504–518. Springer, Heidelberg (2006)
[15] Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam Filtering with Naive Bayes– Which Naive Bayes? In: Third Conference on Email and Anti-Spam (CEAS), pp. 125–134 (2006)
[16] Meyer, T.A., Whateley, B.: SpamBayes: Effective open-source, Bayesian based, email classification system. In: Proceedings of the First Conference on Email and Anti-Spam (CEAS) (2004), http://ceas.cc/papers-2004/136.pdf
[17] Oda, T.: A Spam-Detecting Artificial Immune System. Masters thesis, Carleton University (2005)
[18] Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: Learning for Text Categorization: Papers from the 1998 Workshop, pp. 55–62 (1998)
[19] Tsymbal, A.: The problem of concept drift: definitions and related work. Informe técnico: TCD-CS-2004-15, Departament of Computer Science Trinity College, Dublin, 4(15) (2004)
[20] Yue, X., Abraham, A., Chi, Z., Hao, Y., Mo, H.: Artificial immune system inspired behavior-based anti-spam filter. Soft Computing-A Fusion of Foundations, Methodologies and Applications 11(8), 729–740 (2007)

MOBAIS: A Bayesian Artificial Immune System for Multi-Objective Optimization

Pablo A.D. Castro and Fernando J. Von Zuben

Laboratory of Bioinformatics and Bioinspired Computing - LBiC
Department of Computer Engineering and Industrial Automation - DCA
School of Electrical and Computer Engineering - FEEC
University of Campinas - UNICAMP
PO Box 6101, 13083-970, Campinas, Brazil
{pablo,vonzuben}@dca.fee.unicamp.br

Abstract. Significant progress has been made in theory and design of artificial immune systems (AISs) for solving multi-objective problems accurately. However, an aspect not yet widely addressed by the research reported in the literature is the lack of ability of the AIS to deal effectively with building blocks (high-quality partial solutions coded in the antibody). The available AISs present mechanisms for evolving the population that do not take into account the relationship among the variables of the problem, causing the disruption of these high-quality partial solutions. Recently, we proposed a novel immune-inspired approach for single-objective optimization as an attempt to avoid this drawback. Our proposal replaces the traditional mutation and cloning operators with a probabilistic model, more specifically a Bayesian network representing the joint distribution of promising solutions and, subsequently, uses this model for sampling new solutions. Now, in this paper we extend our methodology for solving multi-objective optimization problems. The proposal, called Multi-Objective Bayesian Artificial Immune System (MOBAIS), was evaluated in the well-known multi-objective Knapsack problem and its performance compares favorably with that produced by contenders such as NSGA-II, MISA, and mBOA.

1 Introduction

A multi-objective optimization problem consists of optimizing a set of conflicting objectives simultaneously. An approach to solve such problems is to consider all objective functions and discover a set of solutions which represents an optimal trade-off between these objectives. This set of solutions is called Pareto optimal set and forms the Pareto front in the space of objectives.

Over the last decades, a variety of evolutionary algorithms have been proposed for solving multi-objective optimization problems, giving origin to the so called Multi-Objective Evolutionary Algorithms (MOEAs). Among the appealing approaches, artificial immune systems (AISs) have received special attention due to their interesting features: (i) dynamical control of population size, in

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 48–59, 2008.

response to the particularities of the problem; (*ii*) efficient mechanism of exploration/exploitation of the search space, which allows to find and preserve the local optima as well as to insert and maintain diversity in the population.

A wide range of AISs have been proposed in the literature for solving multi-objective optimization problems with alternative views of the immune system. One strongly accepted perspective is the one based on both the clonal selection theory [1] and the immune network theory [2]. The first attempt to explore these features of AISs on multi-objective optimization problems was conducted by Yoo and Hajela [3]. However, their approach is coupled with a genetic algorithm (GA). In this case, the immune algorithm is applied only to maintain diversity in the population of the GA. The genuine first AIS for multi-objective optimization, namely Multi-objective Immune System Algorithm (MISA), was proposed by Coello Coello and Cortez [4] and further extended in [5]. Next, Luh et al. proposed the Multi-objective Immune Algorithm (MOIA) [6], Freschi and Repetto proposed the Vector Immune System (VIS) [7], Coelho and Von Zuben proposed the omni-aiNet algorithm [8], and Chen and Mahfouf proposed the PAIA algorithm [9].

Despite their high performance as general problem solving, there are some shortcomings associated with these immune-inspired algorithms. Firstly, as the complexity and scale of the problem increase, the performance of the algorithms becomes more and more associated with a proper choice of the design parameters, such as mutation rate. Otherwise, very poor solutions can be generated. In addition, it is noticeable that, when the solution is represented by a vector of attributes, the population of candidate solutions may contain partial high-quality solutions to the problem, called building blocks. The existing AIS suffer from the lack of ability to identify and effectively manipulate building blocks of the problem. As affinity maturation requires cloning followed by the mutation of the newly-generated cells, and assuming that the mutation operator cannot discover by itself crucial relationships among the variables of the problem, building blocks are not supposed to survive, being disrupted by mutation.

Recently, we proposed an artificial immune system capable of manipulating building blocks effectively, denoted Bayesian Artificial Immune System (BAIS) [10]. Like Estimation of Distribution Algorithms [11] [12] [13], our proposal replaces the traditional mutation operator with a probabilistic model which represents the probability distribution of the promising solutions found so far. Then the obtained probabilistic model is used to generate new individuals. A Bayesian network is adopted here as the probabilistic model, due to its capability of properly capturing the most relevant interactions among the variables of the problem. Since we are replacing the mutation operator, we eliminate the necessity of specifying this parameter value.

Now, we extend the proposal in [10] aiming at investigating its usefulness in multi-objective optimization problems, guiding to Multi-Objective Bayesian Artificial Immune System (MOBAIS). The main objective of this study is not to design an algorithm that produces better results than the state-of-the-art multi-objective evolutionary algorithms reported in the literature. We intend to

design a competent algorithm with qualitative advantages over the contenders, as will be outlined in Section 4. Generally, the quantitative advantages arises as a natural consequence. Experiments on the multi-objective version of the Knapsack problem have been carried out to evaluate the effectiveness of the proposed methodology when compared to other algorithms.

This paper is organized as follows. In Section 2, we provide a background to multi-objective optimization problems. Section 3 describes the MOBAIS in details. The experimental results are outlined and analyzed in Section 4. Finally, in Section 5 we draw some concluding remarks and present the further steps of the research.

2 Multi-Objective Optimization

A multi-objective optimization problem (MOP) is a simultaneous search process for optimal or near optimal trade-off solutions, given some conflicting objective functions. Formally, an MOP consists of minimizing/maximizing the vector function:

$$f(x) = [f_1(x), f_2(x), ..., f_m(x)] \tag{1}$$

subject to J inequality constraints and K equality constraints as follows:

$$g_j(x) \geq 0, \ j = 1, 2, ..., J \tag{2}$$

$$h_k(x) = 0, k = 1, 2, ..., K \tag{3}$$

where $x = [x_1, ..., x_n] \in \Omega$ is the vector of decision variables and Ω is the search space.

When we have a single-optimization objective f, the optimal solution corresponds to the point (or set of points) that has the smallest values of f, considering the whole search space (in a minimization problem). However, for several objective functions, the notion of optimal solution is different, because the aim now is to find good trade-offs among the objective functions. In this case, the most commonly adopted notion of optimality is the one associated with the *Pareto optimality*, which uses the concept of dominance.

Suppose a problem with m objective functions $f_i(x)$, i=1,2,...,m which, without loss of generality, should be minimized. So, we present the following concepts:

1. *Pareto dominance*: a solution x is said to dominate a solution y (denoted by $x \preceq y$) iff $\forall i \in \{1,2,...,m\}$: $f_i(x) \leq f_i(y) \bigwedge \exists i \in \{1,2,...,m\}$: $f_i(x) < f_i(y)$.
2. *Pareto optimal*: a solution x is said to be Pareto optimal iff $\nexists y$: $y \preceq x$.
3. *Pareto optimal set*: is the set PS of all Pareto optimal solutions: $PS = \{x | \ \nexists y : y \preceq x \}$.
4. *Pareto front*: is the set PF of objective function values of all Pareto optimal solutions: PF=$\{\mathrm{F}(x) = [f_1(x),..., f_m(x)] \mid x \in PS\}$.

Notice that the Pareto front consists of diverse trade-off non-dominated solutions in the objective space. Therefore, there are two goals that a multi-objective optimization algorithm must try to achieve: (*i*) guide the search toward the Pareto front; and (*ii*) maintain diverse solutions uniformly distributed along the Pareto front.

3 Multi-Objective Bayesian Artificial Immune System

We propose a novel immune-inspired algorithm which has the mutation and cloning operators replaced by a probabilistic model in order to generate new antibodies. We may interpret our proposal as an Estimation of Distribution Algorithm that adopts an artificial immune system to implement the population-based search strategy and a Bayesian network to implement the probabilistic model, due to its capability of properly representing complex interactions among the variables.

The pseudo-code of the proposed algorithm, called Multi-Objective Bayesian Artificial Immune System (MOBAIS), is presented in Algorithm 1. Notice that the cloning and mutation steps were replaced by the building of the Bayesian network and the subsequent sampling of new individuals according to the generated probabilistic model.

Algorithm 1. Multi-Objective Bayesian Artificial Immune System

```
Begin
  Initialize the population;
  While stopping condition is not met do
    Select the best solutions;
    Build the Bayesian network;
    Sample new individuals;
    Suppress antibodies with fitness lower than a threshold;
    Eliminate similar antibodies;
    Insert new antibodies randomly;
  End while
End
```

In MOBAIS, the initial population is also generated at random. From the current population, the best solutions are selected using a special selection operator, as described in the next subsection. A Bayesian network that better fits the selected antibodies is constructed. A number of new antibodies sampled from the network are inserted into the population. Similar antibodies in the variable space and antibodies with lower fitness are eliminated. Next, a small percentage of individuals are generated randomly and inserted into the population in order to maintain diversity.

Some aspects of MOBAIS should receive special attention. The first one concerns the selection operator. Other aspect is how to suppress similar antibodies.

Finally, the last two stages are related to a way of building the Bayesian network from the selected individuals and how to use the network to generate new solutions. In what follows we explain how to perform these tasks.

3.1 Selection

MOBAIS utilizes a selection operator which ranks the solution based on dominance. This operator is inspired by the selection operator of NSGA-II [14]. First, MOBAIS starts by assign rank 1 to the set of solutions that are not dominated by any other solution in the population. Next, solutions that are not dominated by any of the remaining solutions are assigned rank 2. That is, all solutions with rank 2 are dominated by at least one solution with rank 1, but are not dominated by others in the population. The ranking process continues until all solutions are ranked by assigning increasing ranks to those solutions that are not dominated by any of the remaining.

With respect to Pareto optimality, priority will be given to solutions with lower ranks. Apart from finding solutions closer to the Pareto front, it is also essential to achieve good coverage of the Pareto front. So, a mechanism to maintain diversity in the objective space is desirable. MOBAIS utilizes a mechanism based on crowding distance, which depends on the density of solutions in the neighborhood of each solution. The higher the crowding distance of the solution, the less dense its neighborhood.

3.2 Suppression

In the suppression phase, the Euclidean distance in the variable space among every individual in the population is calculated and normalized with respect to the maximum distance found so far. In this context, the individuals close enough to each other according to a suppression threshold (defined by the user), are subject to a binary tournament procedure and the worst one, in terms of Pareto dominance, is eliminated from the population.

3.3 Bayesian Network - Learning and Sampling

Formally, a Bayesian network for a set of variables $X = \{x_1, x_2, ..., x_n\}$ is a directed acyclic graph whose nodes are variables of the problem and the edges indicate relationships of dependence among the connected variables. Next, we briefly describe how to build a Bayesian network from data and how to use this model to sample new data. In MOBAIS, the Bayesian network learning from a given set of promising solutions corresponds to estimating their joint distribution. Sampling new instances according to the network guides to new candidate solutions to the problem.

Bayesian Network Learning. The Bayesian network learning from a dataset can be stated as follows. Given a collection of observed data, find the network model that explains these data with maximum likelihood. By finding the network we mean to provide the structure of the graph, as well as the probability distribution of each variable that best fits the data.

One usual approach to this task is to adopt a procedure for searching the space of all possible candidate network structures, given a metric that can provide a relative score to each point in the space. Thus, the problem of Bayesian network learning reduces to the problem of searching for a model that yields the highest score, given the observed data. Usually, a heuristic search algorithm is used. It begins with an initial network generated at random. Next, the probability distribution of each variable is estimated using the dataset, and the score of the network is computed. The search process generally proposes small changes to the current structure in order to obtain a network with higher score than the previous one. These small changes can be accomplished by adding or deleting an edge, or reversing the edge direction. Every time a change is made it is necessary to compute the probability distribution of the variables for the modified network. Several algorithms can be used as the search engine [15]. Usually, due to their effectiveness in this context, simple local search based methods are adopted.

Regarding the scoring metrics, there are several measures proposed in the literature. Most of them evaluate a structure S taking into account the likelihood. In this context, a well-known evaluation measure is the so called K2 metric, proposed by Cooper & Herskovits [16]. Given a Bayesian network structure S, and assuming that the data set D is complete (there are no missing values) and that there are no prior knowledge, the likelihood takes the form:

$$p(D|S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{4}$$

where n is the number of instances, q_i denotes the number of possible instances of parents of x_i, r_i is the number of possible values of x_i, N_{ijk} is the number of cases where x_i takes the k-th value with its parents taking their j-th value, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. To avoid round errors during the multiplication of probabilities, often the logarithm is applied to Equation (4).

Sampling in Bayesian Network. Once the Bayesian network is built, we can generate new instances using the joint probability distribution encoded by the network, more specifically $P(X) = \prod_{i=1}^{n} P(x_i|\pi_{x_i})$. To accomplish this task, the Probabilistic Logic Sampling algorithm (PLS) [17] is chosen. PLS finds an ancestral ordering of the nodes in the Bayesian network and instantiates one variable at a time in a forward manner, that is, a variable is not sampled until all its parents have already been sampled.

4 Experimental Results

This section describes the experiments carried out to evaluate the proposed algorithm. We have applied MOBAIS to the well-known multi-objective knapsack problem and compared the performance with other multi-objective optimization tools reported in the literature.

4.1 Multi-Objective Knapsack Problem

The multi-objective Knapsack problem was first used to test MOEAs by Zitzler and Thiele [18]. Consider a set of n items and m knapsacks with a specific capacity. Each item can have a different weight and profit in every knapsack. Selecting item i in a solution implies to put it into every knapsack. A solution cannot exceed the capacity of any knapsack. Formally, the objective is:

$$\begin{aligned} \text{maximize} \quad & f(x) = [f_1(x), f_2(x), ..., f_m(x)] \\ \text{subject to} \quad & \sum_{i=1}^{n} w_{i,j} * x_i \leq c_j \quad , j = 1, ..., m \end{aligned}$$

where x=(x_1, x_2, ..., x_n) $\in \{0,1\}^n$, such that x_i=1 iff item i is packed, $w_{i,j}$ is the weight of item i in knapsack j, $f_j(x) = \sum_{i=1}^{n} p_{i,j} * x_i$, with $p_{i,j}$ being the profit of item i in knapsack j, and c_j is the maximum capacity of knapsack j.

Due to the existence of constraints, a mechanism to deal with them is desired in order to transform infeasible solutions into feasible ones. During the experiments, if a solution violates a constraint, a repair mechanism iteratively removes items until all constraints are satisfied. The order in which the items are checked is determined by the maximum profit/weight ratio. Items with the lowest profit/weight ratio are removed first.

4.2 Experimental Setup

We have applied MOBAIS to the knapsack problem using 2 objectives, varying the number of items. In order to compare the performance of our algorithm and other evolutionary algorithms with known results, we have used two knapsack benchmarks containing 100 items and 250 items, and published on the web site `http://www.tik.ee.ethz.ch/~zitzler/testdata.html`

We generated random weights and random profits in the interval [10,100]. The capacity of a knapsack was set at half of the total weight of all the items: $c_j = 0.5 \sum_{i=1}^{n} w_{i,j}$.

Comparative analysis were carried out taking into account 3 algorithms in the literature. The first one is the well-known NSGA-II [14], that employs non-dominated sorting and crowding distance. The other algorithm is the Multi-objective Immune System Algorithm (MISA), proposed in [4] and which uses a secondary population to implement elitism. Finally, the Multi-objective Bayesian Optimization Algorithm (mBOA) [20], is an Estimation of Distribution-based algorithm that also utilizes a Bayesian network to capture relationships among the variables.

The population size and the number of iterations for each algorithm varies according to the problem, as described in what follows. For NSGA-II algorithm, the crossover and mutation rates were 0.8 and 0.01, respectively, together with tournament selection. In MISA, a uniform mutation was applied to the good solutions and a nonuniform mutation to the “not so good solutions”. MOBAIS

and mBOA have utilized the K2 metric to learn the Bayesian network. In order to penalize the complexity of the model, we have imposed a constraint in the number of parents a node can have. It corresponds to a maximal order of interactions that can be covered and it influences directly the complexity of the model. By our previous experience on Bayesian network learning, we know that when the complexity of the network is too high, it is more likely to detect spurious correlations on the data. Thus, each variable can have only two parents. Once the network is built, we apply the PLS algorithm to generate new individuals. Let N be the size of the current population. So, in MOBAIS the number of samples generated is $N/2$ and in mBOA is N.

These parameters were obtained empirically or referring to the literature and were utilized in all experiments.

4.3 Results

Firstly, we show the obtained results for the Knapsack problem when the number of items (n) is equal to 100. The initial population of MOBAIS was set to 100 and for the other algorithms was set to 300. Since MOBAIS can adjust automatically the population size along the search process, it would not be fair to run all the algorithms with the same population size. The number of iterations for MOBAIS and mBOA was 100. Since these algorithms have the ability to discover and explore the problem regularities, a good Pareto front should be found within a low number of generations. NSGA-II and MISA had the number of iterations set to 300. Figure 1(a) shows the comparison of the Pareto fronts produced by the four algorithms.

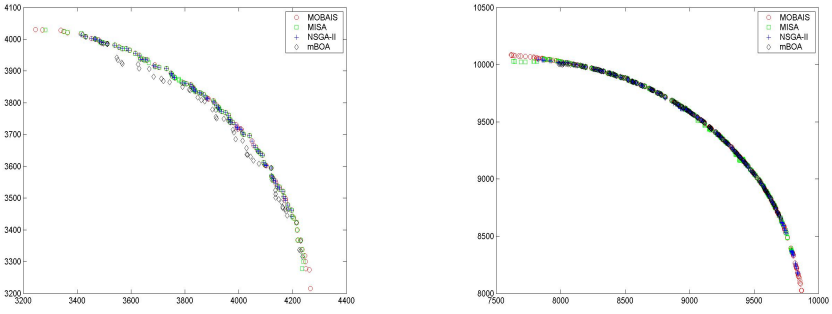
Next, in order to verify the scalability of the algorithms, we have applied them to a larger size problem where the number of items is equal to 250. The number of generations for MOBAIS is still 100 and to the other algorithms is 600. The initial population size for MOBAIS remains 100 and for the other methodologies is 900. For this scenario, the Pareto fronts can be viewed in Figure 1(b).

From Figure 1, we can observe that all algorithms have found a good Pareto front for both scenarios. MOBAIS presented a very good performance because it explores more efficiently the search space using its automatic control of population size and due to its capability to identify and preserve the building blocks. Although mBOA is also able to deal with building blocks, its inferior performance is due to its ineffective mechanism of exploration/exploitation of the search space, when compared with MOBAIS.

In addition to the graphical presentation, we also show in Table 1 the percentage of individuals of an algorithm dominated by the individuals of another algorithm using the coverage metric [21]:

$$C(X', X'') = \frac{|a'' \in X''; \forall a' \in X' : a' \preceq a''|}{|X''|} \tag{5}$$

where $X', X'' \subseteq X$ are two sets of phenotype decision vectors. The output of this metric is a real value in the interval [0,1]. This means that C=1 when X'



(a) (b)

Fig. 1. Pareto fronts produced by MOBAIS, MISA, NSGA-II and mBOA for Knapsack problem with (a) 100 items and (b) 250 items

Table 1. Average values for coverage over 10 executions (A $\preceq$ B)

Algorithm A	Algorithm B	n=100	n=250
MOBAIS	NSGA-II	17%	29%
	MISA	100%	62%
	mBOA	100%	100%
NSGA-II	MOBAIS	8%	13%
	MISA	76%	84%
	mBOA	100%	100%
MISA	MOBAIS	0%	11%
	NSGA-II	12%	5%
	mBOA	100%	83%
mBOA	MOBAIS	0%	0%
	NSGA-II	0%	0%
	MISA	0%	8%

Table 2. Average number of solutions in the Pareto front

Algorithm	n=100	n=250
MOBAIS	**104**	**209**
MISA	72	197
NSGA-II	96	201
mBOA	57	187

dominates or equals X''. Note that both $C(X', X'')$ and $C(X'', X')$ have to be considered, since $C(X', X'')$ is not necessarily equal to 1-$C(X'', X')$.

From Table 1 we can observe that MOBAIS have achieved a good coverage rate over the contenders.

Regarding the number of solutions in the Pareto front, we notice that MOBAIS and NSGA-II have achieved a much broader spread of results than

the other algorithms. Table 2 shows the average number of solutions in the Pareto front over 10 runs, for each algorithm in the two experiments.

4.4 Discussions

As stated in Section 1, MOBAIS offers significant and qualitative advantages over the contenders and they are described below. The first one is related to the effective maintenance of building blocks. With this capability, MOBAIS avoids disrupting the partial solutions found so far. Besides, the replacement of mutation and cloning operators with a probabilistic model eliminates the necessity of defining parameter values for these operators. The same does not occur for the other algorithms. For example, several preliminary experiments were carried out to define adequate values for crossover and mutation operators in NSGA-II.

During the experiments, we also have observed that the preservation of building blocks leads to a quick convergence. While MOBAIS found a good Pareto front in a few generations, the other methodologies needed more generations to achieve the same result. Although mBOA is also able to identify building blocks, its performance was inferior when compared with MOBAIS because MOBAIS has a better mechanism to explore/exploit the search space.

Another advantage of MOBAIS over the contenders is its capability to control the population size in response to the particularities of the problem, allowing a more efficient exploration/exploitation of the search space. Consequently, the initial population size is not crucial to MOBAIS, differently from MISA, NSGA-II and mBOA.

Regarding the implementation of MOBAIS, we notice that the algorithm does not require a large amount of computation resources. The only drawback is the time spent to build the Bayesian network at each iteration. However, the proposed methodology still preserves the computational tractability due to the restriction of at most two parents for each node in the network. The relatively high computational cost to implement MOBAIS is in contraposition with the aforementioned advantages of the algorithm.

Roughly comparing the computational cost of MOBAIS, MISA, NSGA-II and mBOA, in terms of execution time, we could observe that MOBAIS requires much less individuals and much less generations than MISA and NSGA-II, and thus produces a slightly better execution time. When compared with mBOA, the computational burden is equivalent.

5 Conclusion

In this paper we have proposed a novel immune-inspired algorithm for solving multi-objective problems. Our proposal, called Multi-Objective Bayesian Artificial Immune System (MOBAIS), replaces the traditional mutation and cloning operators with a probabilistic model representing the joint distribution of promising solutions and, subsequently, uses this model for sampling new solutions. The probabilistic model used is a Bayesian network due to its capability of properly capturing the most relevant interactions among the variables of

the problem, representing a significant attempt to improve the performance of immune-inspired algorithms when dealing with building blocks.

To evaluate the algorithm, we have applied it to the multi-objective Knapsack problem and compared the obtained results with those produced by state-of-the-art- approaches. All algorithms presented similar performance in terms of coverage of the Pareto front. However, MOBAIS offers qualitative advantages over the contenders such as (*i*) automatic identification/preservation of building blocks, yielding a quick convergence; (*ii*) no necessity of specifying important parameter values to guide the search; (*iii*) adaptive population size in response to the particularities of the problem, allowing a more efficient exploration/exploitation of the search space. Consequently, the initial population size is not crucial to MOBAIS, differently from the compared algorithms.

We are currently investigating some aspects that can be further improved, such as alternative metrics for evaluating the Bayesian networks and other algorithms for sampling new individuals. We are also analyzing the performance of MOBAIS in other problems. Another aspect to be considered is the extension of the proposal to handle optimization problems in a continuous domain.

Acknowledgments

The authors would like to thank the Brazilian Research Council (CNPq) for the financial support.

References

1. Ada, G.L., Nossal, G.J.V.: The Clonal Selection Theory. Scientific American 257(2), 50–57 (1987)
2. Jerne, N.K.: Towards a Network Theory of the Immune System. Ann. Immunol (Inst. Pasteur) 125C, 373–389 (1974)
3. Yoo, J., Hajela, P.: Immune network simulations in multicriterion design. Structural Optimization 18, 85–94 (1999)
4. Coello Coello, C., Cortés, N.C.: An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune System. In: First International Conference on Artificial Immune System, pp. 212–221 (2002)
5. Coello Coello, C., Cortés, N.C.: Solving Multiobjective Optimization Problems Using an Artificial Immune System. Genetic Programming and Evolvable Machines 6(2), 163–190 (2005)
6. Luh, G.-C., Chueh, C.-H., Liu, W.-M.: MOIA: Multi-objective Immune Algorithm. Engineering Optimization 35(2), 143–164 (2003)
7. Freschi, F., Repetto, M.: VIS: An artificial immune network for multi-objective optimization. Engineering Optimization 38, 975–996 (2006)
8. Coelho, G.P., Von Zuben, F.J.: Omni-aiNet: An Immune-Inspired Approach for Omni Optimization. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 294–308. Springer, Heidelberg (2006)
9. Chen, J., Mahfouf, M.: A Population Adaptive Based Immune Algorithm for Solving Multi-objective Optimization Problems. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 280–293. Springer, Heidelberg (2006)

10. Castro, P.A.D., Von Zuben, F.J.: BAIS: A Bayesian Artificial Immune System for the Effective Handling of Building Blocks. Information Sciences - Special Issue on Artificial Immune System (accepted, 2008)
11. Mühlenbein, H., Paass, G.: From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. In: 4th Int. Conf. on Parallel Problem Solving from Nature, pp. 178–187 (1996)
12. Baluja, S.: Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Technical Report, Carnegie Mellon University, Pittsburgh, PA, USA (1994)
13. Pelikan, M., Goldberg, D., Lobo, F.: A survey of optimization by building and using probabilistic models, Technical Report, University of Illinois, ILLIGAL Report n 99018 (1999)
14. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation 6(2), 182–197 (2002)
15. Castro, P.A.D., Von Zuben, F.J.: Bayesian Learning of Neural Networks by Means of Artificial Immune Systems. In: 5th Int. Joint Conf. on Neural Networks, pp. 9885–9892 (2006)
16. Cooper, G., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning 9, 309–347 (1992)
17. Henrion, M.: Propagating uncertainty in Bayesian networks by probabilistic logic sampling. Uncertainty in Artificial Intelligence 2, 149–163 (1998)
18. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)
19. Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations, PhD Thesis, Graduate School of Engineering of the Air Force Inst. of Tech., Wright-Patterson AFB (1999)
20. Khan, N., Goldberg, D.E., Pelikan, M.: Multi-Objective Bayesian Optimization Algorithm, Illigal Report 2002009 (2002)
21. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2), 173–195 (2000)

An Advanced Clonal Selection Algorithm with Ad-Hoc Network-Based Hypermutation Operators for Synthesis of Topology and Sizing of Analog Electrical Circuits

Angelo Ciccazzo[1], Piero Conca[2], Giuseppe Nicosia[2], and Giovanni Stracquadanio[2]

[1] ST Microelectronics
Stradale Primosole 50, 95121 Catania, Italy
{angelo.ciccazzo}@st.com
[2] Department of Mathematics and Computer Science
University of Catania
Viale A. Doria 6, 95125 Catania, Italy
{conca,nicosia,stracquadanio}@dmi.unict.it

Abstract. In electronics, there are two major classes of circuits, analog and digital electrical circuits. While digital circuits use discrete voltage levels, analog circuits use a continuous range of voltage. The synthesis of analog circuits is known to be a complex optimization task, due to the continuous behaviour of the output and the lack of automatic design tools; actually, the design process is almost entirely demanded to the engineers. In this research work, we introduce a new clonal selection algorithm, the *elitist Immune Programming*, (eIP) which uses a new class of hypermutation operators and a network-based coding. The eIP algorithm is designed for the synthesis of topology and sizing of analog electrical circuits; in particular, it has been used for the design of passive filters. To assess the effectiveness of the designed algorithm, the obtained results have been compared with the passive filter discovered by Koza and co-authors using the Genetic Programming (GP) algorithm. The circuits obtained by eIP algorithm are better than the one found by GP in terms of frequency response and number of components required to build it.

1 Introduction

The immune system consists of a complex network of process interactions, which cooperates and competes to contrast the antigen attacks. Theory of clonal selection principle hypothesizes that B-cells contrast the infections by means of a series of measures. Every being has a very large population of different B-cells within its body. In case an external entity, such as a virus or a bacterium, trespasses the body barriers, B-cells start trying to match the external body or antigen, by means of the receptors present on their cell surface. When the receptors of a B-cell totally or partially match the antigen, the B-cell starts to

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 60–70, 2008.

proliferate in a process called *clonal expansion*. Moreover, the cloned B-cells can undergo to somatic mutations, in order to increase the affinity with an antigen: it is a Darwinian process of variation and selection, called *affinity maturation* [1]. This bottom-up behaviour has received a great attention in computer science, and it is the main source of inspiration for the emerging class of *Immune Algorithms* [2,3,4,5,6].

In electronics, the design of analog circuits is an iterative process accomplished by skilled engineers. There is no CAD tool that automatically designs analog circuits starting from a set of requirements [7]. The main idea is to find a general methodology that makes effective this working flow in order to automatically design new analog circuits and speeding up the time-to-market for new devices [8,9]. In order to tackle this problem, the *elitist Immune Programming* algorithm (EIP) is introduced: it extends the *Immune Programming* (IP) algorithm [10] with the introduction of *elitism* and ad-hoc hypermutation operators for handling analog circuits. The EIP algorithm is adopted for the design of analog circuits belonging to the class of *passive filters*. A Passive filter is an interesting test-bed tackled firstly by the Genetic Programming (GP) algorithm [11,12,13,14]. We have conducted several experiments in order to highlight two important aspects: firstly, how the elitism impacts the exploring and exploiting ability of the immune programming algorithm; secondly, the suitability of EIP for the automatic synthesis and sizing of analog electrical circuits. The obtained experimental results confirm that EIP outperforms the standard IP approach in terms of convergence speed and quality of the designed circuits; moreover, the new immune algorithm is able to design passive filters that are clearly better than the one discovered using GP in terms of frequency response and number of components required.

In section two we give an overview on the passive filters; in section three we describe the *elitist Immune Programming* algorithm; in section four, we report our experimental results and in section five we outline conclusions and future works.

2 Passive Filters Circuits

Passive filters are a particular class of analog circuits, which are made of passive components, such as resistors, capacitors and inductors. Given a signal, a filter leaves it unchanged in a frequency range called *pass band*, instead, in a frequency range called *stop band*, it attenuates the signal below a certain level. In the pass band a *ripple voltage* (V_r) should be achieved; V_r is the maximum acceptable amplitude of the oscillations in pass band. In the range between pass and stop bands, called *transition band*, the filter must reduce the input signal amplitude in order to reach the desired attenuation with a very smooth behaviour. Slight deviations from an ideal behaviour are considered acceptable and they are specified by the two deviation parameters d and h.

The circuit contains a test structure and a circuit core, in this way, the same operating conditions are used for every circuit put into the core structure. The test structure is made of a signal generator (VSOURCE), a series resistance

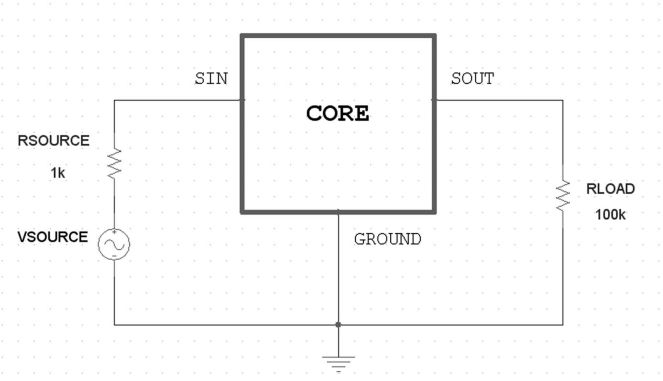


Fig. 1. Passive Filter Circuit. It is possible to note the shunt resistance, RSOURCE, the Load Resistance, RLOAD and the power supply source, VSOURCE.

(RSOURCE), a Load Resistance (RLOAD) and a Ground link. This structure supplies three links, the first link provides the power voltage to the circuit core, which is connected to the load resistor via the second link and the third provides the connection to the ground, as shown in Fig. 1.

In our experiments, we synthesize a passive filter with a cut-off frequency of $1KHz$ and a transition band of $1KHz$. The value for d and h were settled respectively at $0.1V$ and $10^{-4}V$ and the V_r parameter was settled to $0.03V$. The set of available values for resistors and capacitors is that of the commercial series E-24. The order of magnitude of resistors values ranges from $10^{8}\Omega$ to $10^{-2}\Omega$, while the order of magnitude of capacitors ranges from $10^{-1}F$ to $10^{-11}F$. For inductors there is not an analogue standardization, so we have chosen values ranging from $1H$ to $10^{-11}H$ with a step size of 0.1 [12].

3 Immune Programming for Analog Circuit Design

In this section we give an overview of the standard IP algorithm and, successively, we give a detailed description of the new EIP algorithm.

3.1 Immune Programming

The *Immune Programming.* (IP) is a population-based algorithm inspired by the clonal selection principle. The algorithm starts with a population of randomly generated B-cell. At each generation g, IP builds a new population by considering each B-cell for replacing, cloning or hypermutation. The process is iteratively performed until the maximum number of generations (or objective function evaluations) is reached. The *Replacement* operator replaces a B-cell of the population with a new random one, it is mainly employed in the early stage of the evolutionary process, and it is one of the major responsible for the diversity of the current population. The *Cloning* operator is used to create multiple

copies of the best individuals in the population, this operator gives more chance to explore a promising region of the solution space. The *Hypermutation* operator is used to modify a B-cell according to its fitness value and it is the crucial point for the exploring ability of the algorithm.

In the IP algorithm, these three operators are controlled by three parameters P_r, P_c, P_m. P_r represents the minimum percentage of newly generated B-cell at each iteration, and it is inversely proportional to the average fitness function value of the previous generation. In the early stage of the evolutionary process, the IP algorithm makes a lot of replacements that decrease when a good repertoire of solutions is established. The parameter P_c controls the ratio between number of cloned B-cells and the number of B-cells that will be mutated. The last parameter P_m represents the percentage of receptors of the best B-cell that will be mutated; according to this strategy, the best circuit is less mutated than the worst ones that can undergo a complete mutation of each receptor.

3.2 The Elitist Immune Programming Algorithm

IP was the starting point to develop the new *elitist Immune Programming* (EIP) algorithm; the pseudo-code of the algorithm is provided in Fig.2. EIP differs from IP in several points, the following new features are introduced to effectively tackle the synthesis of topology and the sizing of analog circuits.

Firstly, the algorithm was modified with the introduction of *elitism*. At each generation g, the best solution found so far cannot be erased from the population. This strategy, already introduced in other *immune inspired* algorithms [5,6,2], greatly helps the convergence of the algorithm and it overcomes the problem of IP that tends to quickly forget good solutions especially in the initial phase of the search process. The other main difference is the application of the cloning and hypermutation operators. As in IP the chance to be cloned or mutated is driven by a parameter P_c but, in EIP, for each cloning two mutations are performed.

Mutation Operators. The hypermutation operators operate only on the core structure; in particular, the hypermutation acts on one component, link or node at a time. All the operators take in input and return in output only consistent circuits. This design choice forces the algorithm to search in the feasible region of the solution space, and it helps the algorithm to immediately discard infeasible or meaningless solutions. Ten different mutation operators have been introduced, and each of them makes a specific mutation on the circuit as described below.

`ADD-SERIES`. Given a circuit, it randomly selects a component and it randomly unplugs one of its terminals; successively, a new component is created and the operator connects it in series to the selected component, linking the floating terminal to the new one.

`ADD-PARALLEL`. It establishes a shunt connection. After a component is selected, the operator randomly creates a new component and then it links its terminals to the same nodes of the selected one.

```
1: procedure EIP(D, MaxGen, P_r, P_m)
2:     G ← 1
3:     Population^(0) ← Initialize(D)
4:     Evaluate(Population)
5:     while G < MaxGen do
6:         Population^(G+1) ← empty
7:         Population^(G+1) ← BestCircuit[Population^(G)]
8:         Population^(G+1) ← Hypermutation[BestCircuit[Population^(G)]]
9:         i ← 0
10:        repeat
11:            if rand() < P_r then
12:                NewCircuit ← Initialize()
13:                Population^(G+1) ← NewCircuit()
14:            else
15:                if rand() < P_c(Circuit_i) then
16:                    Population^(G+1) ← Population_i^(G)
17:                end if
18:                for j ← 1 to 2 do
19:                    if rand() < P_m(Circuit_i) then
20:                        Population^(G+1) ← Hypermutation[Population_i^(G)]
21:                    end if
22:                end for
23:                i ← i + 1 mod D
24:            end if
25:        until size[Population^(G+1)] < D
26:    end while
27: end procedure
```

Fig. 2. The pseudo-code of the EIP algorithm

`ADD-RANDOM-COMPONENT.` It randomly creates a new component that will be connected to two random nodes of the circuit.

`EXPAND-NODE.` This operator randomly selects a circuit node and it randomly generates a new node and a new component. Successively, it connects the new component to the previous selected node.The scope of this procedure is to easily plug in a new component into a highly linked node, or a test structure node.

`DELETE-COMPONENT.` This procedure tries to decrease the size of the circuit by deleting a component. It does not affect the consistency of the circuit; however, if a deletion causes damages, the operator is able to repair the circuit.An inconsistent circuit can arise due to one or more floating terminals, the unplugging of the circuit core from the test structure or the unlinking of a part of the circuit.

`MUTATE-COMPONENT-VALUE.` The operator randomly selects a component and it changes its value by randomly picking a new value from the set of allowed values.

`COPY-COMPONENT-VALUE.` The operator randomly selects a component of the circuit and it copies the value of a randomly chosen component of the same type. If there is no other similar component, it does nothing.

`MUTATE-COMPONENT-KIND.` This operator randomly selects a component, then it modifies the relative type and it assigns a value to the component according to the allowed set of values for the new type.

`LINK-MODIFY.` The operator randomly disconnects a link of a component and reconnects it to a different circuit node. Like `DELETE-COMPONENT`, this procedure is able to recover from inconsistent circuits.

`SHRINK.` The `SHRINK` operator scans the circuit in order to find a series or parallel connection between two or more components. It reduces the circuit size by replacing a couple of components with one equivalent component which value is as close as possible to the values of the two components.This operator greatly improves the quality of the design since it allows the automatic introduction of standard components and the reduction of the circuit size with only marginal side effects [15].

Fitness Function. The quality of the circuit is assessed by means of an ad-hoc objective function; it measures the distance between the curve described by a circuit and the one described by a hypothetical ideal circuit according to the following expression:

$$f_{pf}(x) = \sum_{i=100mHz}^{1KHz} [W_p((f_i), f_i) \times d(f_i)] + \sum_{i=2KHz}^{100MHz} [W_s((f_i), f_i) \times d(f_i)] \qquad (1)$$

where x is a consistent circuit, f_i is the $i-th$ frequency, $d(f_i)$ is the signal deviation from an ideal behaviour and $W_p(d(f_i), f_i)$ and $W_s(d(f_i), f_i)$ are weighting factors respectively for the pass and stop band. For each frequency, the corresponding weighting factor for the pass band is determined as follows:

$$W_p = \begin{cases} 0 & d(f_i) \leq V_r \\ c & V_r < d(f_i) \leq d \\ 10 \cdot c & d(f_i) > d \end{cases}$$

where V_r is the ripple voltage and d, c are experimentally obtained constants that were fixed to $d = 0.1V$ and $c = 3$. The weighting factor for the stop band term is obtained as follows:

$$W_s = \begin{cases} 0 & d(f_i) \leq SBA \\ m & SBA < d(f_i) \leq h \\ 10 \cdot m & d(f_i) > h \end{cases}$$

where SBA is the desired *Stop Band Attenuation*, that was fixed to $-60dB$ and d, h, m are experimentally obtained constants fixed to $d = 0.1V$, $h = 10E - 5V$ and $m = 50$. It is possible to observe that the co-domain of the distance function is $[0, +\infty[$, where an ideal circuit has $f_{pf}(x) = 0$. This distance function neglects small deviations from an ideal behaviour and it strongly penalizes unacceptable deviations. The fitness of each B-cell is the value of f_{pf} normalized in the range $[0, 1]$ according to the following expression:

Table 1. Experimental results, the performances of the two immune algorithms. For each parameters setting, we report the *Circuit with the Lowest Fitness Function value* (CLFF) and *the Circuit with the Lowest Number of Components* (CLNC).

			CLFF		CLNC	
ALGORITHM	d	P_m	f_{pf}	Components	f_{pf}	Components
IP	5×10^3	0.1	1632.04	5	1632.04	5
IP	5×10^3	0.3	1343.03	5	1343.03	5
IP	10^4	0.1	1758.54	3	1758.54	3
IP	10^4	0.3	1742.77	6	1763.77	4
EIP	5×10^3	0.1	20.5486	20	20.948	18
EIP	5×10^3	0.3	10.2221	20	11.3294	16
EIP	10^4	0.1	**0.0**	12	0.29	**10**
EIP	10^4	0.3	8.7778	18	8.78324	16

$$fitness(x_i^g) = \frac{1 - s_{f_{pf}}(x_i^g) \times m_{f_{pf}}(x_i^g)k}{\alpha} \tag{2}$$

$$s_{f_{pf}}(x_i^g) = \frac{f_{pf}(x_i^g)}{f_{pf}^{MAX}(g)} \tag{3}$$

$$m_{f_{pf}}(x_i^g) = e^{\frac{f_{pf}(x_i^g)}{k}} \tag{4}$$

where x_i^g is the $i-th$ B-cell of the population at generation g, $f_{pf}^{MAX}(G)$ is the max value of the objective function at generation g, instead k is a constant used to constraint the fitness in the range $[0, 1]$. Moreover, the fitness was scaled of $\alpha = 25\%$ in order to prevent that the worst B-cell undergoes to a complete mutation of the circuit.

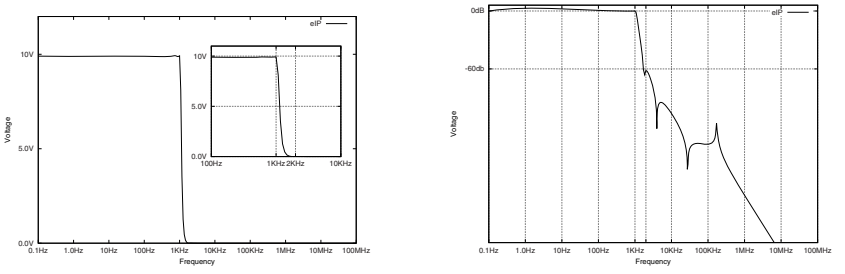
4 Experimental Results

In order to assess the effectiveness of the EIP algorithm, we performed several experiments. Firstly, we compared EIP with the standard IP algorithm. We have tested these two algorithms with a population of size $d \in \{5000, 10000\}$ [16]. The mutation probability parameter was settled to $P_m \in \{0.1, 0.3\}$; since P_m is the percentage of receptor mutated in the best circuit, a larger value of this parameter makes the algorithm acting as a *random search*. Finally, the replacement probability P_r and the cloning probability P_c are fixed to $P_r = 0.01, P_c = 0.2$ [10]. In order to simulate the behaviour of the circuits, the tested algorithms use the NGSPICE circuit simulator. The maximum number of objective function evaluations was set to 10^7 for all the experiments and the same set of mutation operators were used in both algorithms.

It is possible to note in Tab.1,2 that EIP clearly outperforms the IP algorithm. For all settings, EIP shows a good convergence to near optimal solutions, instead IP produces only meaningless circuits. The scheme adopted by IP for replacement, cloning and hypermutation is not effective for this problem; at each iteration only replacement are performed and it means that IP works most likely a random search.

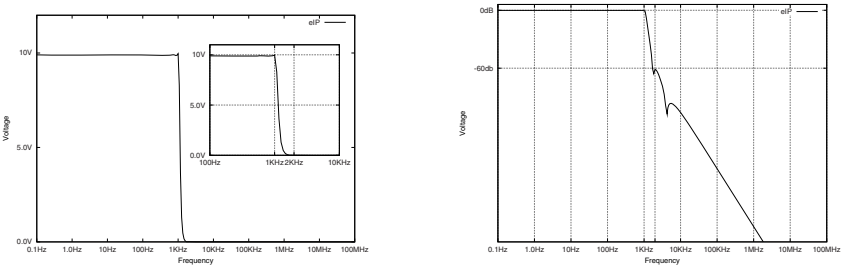
Table 2. Experimental results, a comparison of 5 independent runs of IP and EIP using the best parameter setting according to Tab.1

RUN	ALGORITHM	f_{pf}	COMPONENTS	ALGORITHM	f_{pf}	COMPONENTS
1	IP	1542.72	3	EIP	3.93	20
2	IP	1765.63	6	EIP	16.79	20
3	IP	1658.13	6	EIP	12.62	20
4	IP	1492.22	4	EIP	0.29	10
5	IP	1497.31	3	EIP	0.0	12
	AVERAGE	1591.202	4.4	AVERAGE	6.726	16.4



(a) (b)

Fig. 3. The *output voltage frequency response* (a) and the *attenuation plot* (b) of the best circuit found by EIP ($f_{pf} = 0.0$, 12 components)

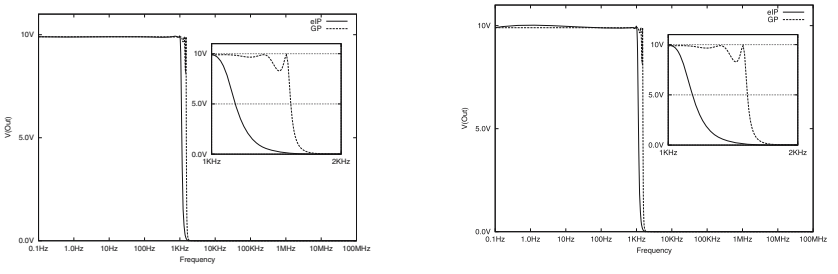


(a) (b)

Fig. 4. The *output voltage frequency response* (a) and the *attenuation plot* (b) of the circuit with the lowest number of components found by EIP ($f_{pf} = 0.29$, 10 components)

By inspecting the EIP results, it is possible to note that using a population of 10000 B-cells and a mutation probability $P_m = 0.1$, the algorithm found a circuit that perfectly matches the design requirements (Fig.3). By analyzing the circuit structure it is possible to note that is made of only 12 components that is an important aspect for the manufacturability of the filter. Moreover, by inspecting all the circuits designed by EIP, the algorithm has found a circuit of 10 components with $f_{pf} = 0.29$ (Fig.4); despite the value of the fitness function

is not optimal, the circuit shows a very regular behaviour and, probably, it can be considered a good trade-off between the requirements and the manufacturability of the filter. By observing the circuits it is possible to note that they show different shapes but common building blocks: this behaviour suggests that EIP is able to find a common regular structure and, at the same time, it is able to arrange them in order to deeply explore the space of solutions. Finally, the *population-based* approach gives to the engineers not only a single solution but a set of circuits that could be inspected in order to find the one that optimally fits the design requirements.



(a) (b)

Fig. 5. A comparison of the output voltage frequency response of the circuit with the optimal fitness function value (a, $f_{pf} = 0.0$, 12 components) and the one with the lowest number of components (b, $f_{pf} = 0.29$, 10 components) found by EIP with the *Campbell filter* [11]. it is possible to note that in the transition band the *Campbell filter* has not a regular behaviour instead the EIP circuits have a regular and smooth curve.

The GP algorithm was able to find a passive filter, known as *the Campbell filter* [11]. This filter shows a very regular structure and a good symmetry, since it is built using the same building block repeated multiple times in order to form a seven rung ladder structure. The frequency response of the Campbell filter is substantially linear in pass band and the curve inclination is very high. The two best circuits found by EIP are better then the Campbell filter for three important aspects. Firstly, in the transition band, the signal of Campbell filter shows large swings that are an undesirable behaviour instead, the EIP circuits show a very regular and smooth curve as showed in Fig.5. Secondly, the EIP circuits have only 10 and 12 components instead the Koza's circuit has 14 components, and this fact makes the EIP circuits more suitable for a real implementation. Finally, the EIP algorithm requires 10^7 fitness function evaluations to design these circuits instead the GP algorithm requires 1.5×10^7 evaluations; this experimental result proves that the immune algorithm, for this design problem, is more efficient than GP.

5 Conclusions and Future Works

In this research work, we have introduced a new *immune algorithm*, called ELITIST IP, for the synthesis of topology and sizing of analog electrical circuits.

The algorithm extends the IMMUNE PROGRAMMING approach with the introduction of *elitism* and *ad-hoc operators* for handling analog circuits.

The experimental results confirms that EIP clearly outperforms the standard IMMUNE PROGRAMMING approach in terms of quality of the circuits and speed of convergence. The analysis of the EIP circuits shows that the algorithm is able to synthesize analog circuits with excellent frequency responses, having small swings, high inclination and a good shape regularity.

The comparison with the *Campbell filter*, a passive filter discovered using GENETIC PROGRAMMING, shows that EIP is able to find a better circuit in terms of regularity in transition band and number of components required.

Starting from these results, there are two major fields that we are investigating. Firstly, we are extending the EIP algorithm in order to use a selection strategy based on the *Pareto Optimality* criterion; using this approach, it is possible to explicitly introduce different design requirements, such as the number of components and the frequency response, and leaving to the algorithm the automatic discovering of optimal trade-off [17]. Finally, we are designing an improved EIP that is able to synthesize the topology and the sizing of active filters [18]; this last task is a visionary research topic since there is not an automatic approach for the design of these analog circuits and it could be an important step to dramatically decrease the time-to-market required for these circuits.

References

1. Abbas, A., Lichtman, A., Pober, J., et al.: Cellular and molecular immunology. WB Saunders, Philadelphia (2000)
2. Cutello, V., Nicosia, G., Pavone, M.: Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator. In: Proceedings of the 2006 ACM symposium on Applied computing, pp. 950–954 (2006)
3. Cutello, V., Nicosia, G., Pavone, M., Timmis, J.: An Immune Algorithm for Protein Structure Prediction on Lattice Models. IEEE Transactions on Evolutionary Computation 11(1), 101–117 (2007)
4. Freitas, A., Timmis, J.: Revisiting the Foundations of Artificial Immune Systems: A Problem-Oriented Perspective. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787. Springer, Heidelberg (2003)
5. Cutello, V., Nicosia, G., Pavone, M.: A hybrid immune algorithm with information gain for the graph coloring problem. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 171–182. Springer, Heidelberg (2003)
6. Cutello, V., Morelli, G., Nicosia, G., Pavone, M.: Immune algorithms with aging operators for the string folding problem and the protein folding problem. In: Raidl, G.R., Gottlieb, J. (eds.) EvoCOP 2005. LNCS, vol. 3448, pp. 80–90. Springer, Heidelberg (2005)
7. Streeter, M., Keane, M., Koza, J.: Iterative Refinement Of Computational Circuits Using Genetic Programming. In: Proceedings of the Genetic and Evolutionary Computation Conference table of contents, pp. 877–884 (2002)

8. Koza, J., Bennett III, F., Andre, D., Keane, M., Dunlap, F.: Automated synthesis of analog electrical circuits by means of genetic programming. IEEE Transactions on Evolutionary Computation 1(2), 109–128 (1997)
9. Kashtan, N., Alon, U.: Spontaneous evolution of modularity and network motifs. Proceedings of the National Academy of Sciences 102(39), 13773–13778 (2005)
10. Musilek, P., Lau, A., Reformat, M., Wyard-Scott, L.: Immune programming. Information Sciences 176(8), 972–1002 (2006)
11. Koza, J., Bennett III, F., Andre, D., Keane, M.: Synthesis of topology and sizing of analog electrical circuits by means of genetic programming. Computer Methods in Applied Mechanics and Engineering 186(2-4), 459–482 (2000)
12. Koza, J., Jones, L., Keane, M., Streeter, M.: Towards industrial strength automated design of analog electrical circuits by means of genetic programming. Genetic Programming Theory and Practice II (2004)
13. Grimbleby, J.: Automatic analogue circuit synthesis using genetic algorithms. Circuits, Devices and Systems. IEE Proceedings [see also IEE Proceedings G-Circuits, Devices and Systems] 147(6), 319–323 (2000)
14. Alpaydin, G., Balkir, S., Dundar, G.: An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. IEEE Transactions on Evolutionary Computation 7(3), 240–252 (2003)
15. Dastidar, T., Chakrabarti, P., Ray, P.: A Synthesis System for Analog Circuits Based on Evolutionary Search and Topological Reuse. IEEE Transactions on Evolutionary Computation 9(2), 211–224 (2005)
16. Koza, J.: Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann, San Francisco (1999)
17. Subramanian, A., Sayed, A.: Multiobjective filter design for uncertain stochastic time-delay systems. IEEE Transactions on Automatic Control 49(1), 149–154 (2004)
18. El-Habrouk, M., Darwish, M., Mehta, P.: Active power filters: a review. IEE Proceedings Electric Power Applications 147(5), 403–413 (2000)

A Multi-Objective Multipopulation Approach for Biclustering

Guilherme Palermo Coelho, Fabrício Olivetti de França, and Fernando J. Von Zuben

Laboratory of Bioinformatics and Bioinspired Computing (LBiC)
Department of Computer Engineering and Industrial Automation (DCA)
School of Electrical and Computer Engineering (FEEC)
University of Campinas (Unicamp)
P.O. Box 6101 - Zip Code 13083-970 - Campinas, SP, Brazil
{gcoelho,olivetti,vonzuben}@dca.fee.unicamp.br

Abstract. Biclustering is a technique developed to allow simultaneous clustering of rows and columns of a dataset. This might be useful to extract more accurate information from sparse datasets and to avoid some of the drawbacks presented by standard clustering techniques, such as their impossibility of finding correlating data under a subset of features. Given that biclustering requires the optimization of two conflicting objectives (residue and volume) and that multiple independent solutions are desirable as the outcome, a multi-objective artificial immune system capable of performing a multipopulation search, named MOM-aiNet, will be proposed in this paper. To illustrate the capabilities of this novel algorithm, MOM-aiNet was applied to the extraction of biclusters from two datasets, one taken from a well-known gene expression problem and the other from a collaborative filtering application. A comparative analysis has also been accomplished, with the obtained results being confronted with the ones produced by two popular biclustering algorithms from the literature (FLOC and CC) and also by another immune-inspired approach for biclustering (BIC-aiNet).

Keywords: biclustering, multi-objective optimization, multipopulation search, artificial immune systems, gene expression, collaborative filtering.

1 Introduction

Due to the increasing amount of information acquired in business, science, internet and biomolecular research, data clustering has become an even more essential subject on knowledge extraction. Classical data clustering tools, such as k-means, Self Organized Maps (SOMs) and Hierarchical Clustering have been successfully applied to different kinds of problems, but they present some limitations when dealing with large and heterogeneous datasets, structured as data matrices of objects (rows) and their corresponding attributes (columns). When dealing with such matrices, these clustering approaches cannot detect partial matching since the dataset is grouped based solely on global similarities (considering all

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 71–82, 2008.

the attributes simultaneously). Also, most of these techniques are only capable of assigning a given object to only one group (cluster), what may be insufficient in several applications, ranging from text mining to complex networks in biology [1].

In order to avoid these drawbacks and knowing that the discovery of similarities between two objects, considering different subsets of attributes, may be useful to make deeper inferences, the *biclustering* technique was proposed [2]. This technique is capable of finding several subsets of rows and columns from the data matrix. In this way, each subset will be composed of objects (rows) that share some similarities specifically on the selected attributes (columns). That is why a single object may take part in multiple biclusters, in association with a distinct subset of attributes at each bicluster, thus allowing the extraction of additional information from the dataset. The problem of finding several biclusters may be considered similar to the problem of finding several two-way bipartitions of the whole dataset, which is clearly a combinatorial optimization problem. Also, the construction of a high-quality bicluster requires a compromise between two conflicting objectives: both the volume of the bicluster and the degree of similarity among its elements should be maximized.

Since the amount of biclusters that can be extracted from a given dataset is previously unknown and due to the multi-objective nature of the problem, an algorithm that performs multi-objective optimization and adopts multipopulation search is likely to be successful in the biclustering generation task.

In 2001, de Castro & Von Zuben [3] have developed the first tool of a family of immune inspired algorithms, called Artificial Immune Network (*aiNet*), that evolves multiple subpopulations in parallel. Given this multipopulation property of aiNet and the importance of the biclustering technique, in this work a multi-objective immune-inspired biclustering algorithm (named *MOM-aiNet*, Multi-Objective Multipopulation Artificial Immune Network) is proposed.

The MOM-aiNet algorithm was applied to two important problems with distinct characteristics: the *Yeast* problem [4], which is a gene expression dataset that has been extensively studied along the biclustering literature, and the *Movielens* dataset [5], which is a set of movie ratings given by the clients of a video rental store. The Movielens problem is considered a challenge to data mining due to its sparseness and the need to correlate a given client to more than one group simultaneously. The results obtained by MOM-aiNet were compared to those of three other algorithms from the literature: the algorithm of Cheng & Church (CC) [2], FLOC (*FLexible Overlapped biClustering* - [6]) and BIC-aiNet (*Artificial Immune Network for Biclustering* - [7], [8], [9]).

This paper is organized as follows. Section 2 presents some general aspects of biclustering. Section 3 outlines the algorithm proposed in this work and the basic immune concepts employed. The experiments performed and the comparison of MOM-aiNet with CC, FLOC and BIC-aiNet are depicted in Section 4. Finally, the concluding remarks of the paper and further steps of the research are presented in Section 5.

2 Biclustering

In data mining, biclustering is referred to the process of finding subsets of rows and columns of a given data matrix [2] (see Fig. 1). This data matrix may represent different kinds of numerical data, such as objects and their attributes (comprising the rows and columns of the matrix, respectively).

The biclustering approach covers a wide scope of different applications, and some examples are dimensionality reduction [10], information retrieval and text mining ([7], [11], [12]), electoral data analysis [13], collaborative filtering ([8], [9], [14]) and biological data analysis ([10], [15]).

The biclustering task can be classified into several categories, according to (*i*) the way the bicluster quality is measured; (*ii*) how the set of biclusters are built; and (*iii*) which structure of bicluster is adopted [16].

The classification based on the quality measure of a biclustering algorithm is related to the concept of similarity between the elements of the matrix. For instance, some algorithms search for constant value biclusters, some for constant columns or rows, and others for coherency in the values of the elements. In Fig. 1 some of the quality measures of biclustering algorithms are illustrated. Of course, in practical applications, the obtained biclusters will not follow the quality measure without some deviation, interpreted as an error (residue) to be minimized at the same time that the volume of the biclusters (to be defined in what follows) is maximized.

$$\underset{\text{(a)}}{\begin{bmatrix} 3 & 1 & 2 & 1 & 5 \\ 4 & 1 & 5 & 1 & 5 \\ 1 & 1 & 3 & 2 & 5 \\ 4 & 2 & 3 & 2 & 6 \\ 5 & 3 & 4 & 3 & 7 \end{bmatrix}} \quad \underset{\text{(b)}}{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}} \quad \underset{\text{(c)}}{\begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix}} \quad \underset{\text{(d)}}{\begin{bmatrix} 1 & 5 \\ 1 & 5 \\ 1 & 5 \end{bmatrix}} \quad \underset{\text{(e)}}{\begin{bmatrix} 3 & 2 & 1 & 5 \\ 4 & 3 & 2 & 6 \\ 5 & 4 & 3 & 7 \end{bmatrix}}$$

Fig. 1. A concise and didactic example of four biclusters ((b), (c), (d) and (e)), each one obeying a specific optimization criterion, extracted from the original matrix (a). The bicluster (b) was created with rows $\{1, 2\}$ and columns $\{2, 4\}$, and is an example of a constant bicluster. The bicluster (c) was created with rows $\{1, 4, 5\}$ and columns $\{2, 4\}$, and is an example of a bicluster with constant rows. The bicluster (d) was created with rows $\{1, 2, 3\}$ and columns $\{2, 5\}$, and is an example of a bicluster with constant columns. The bicluster (e) was created with rows $\{1, 4, 5\}$ and columns $\{1, 3, 4, 5\}$, and is an example of a bicluster with coherent values.

In this paper, biclustering will be employed to find coherence inside biological data on microarray experiments and to extract overlapping information on a sparse dataset used for collaborative filtering. Both applications involve the search for biclusters equivalent to bicluster (e) in Fig. 1.

The way the biclusters are built refers to the number of biclusters discovered per run. Some algorithms find only one bicluster at each run, while others are capable of simultaneously finding several biclusters. Besides, there are nondeterministic and deterministic algorithms. Non-deterministic algorithms are able to

find different solutions for the same problem at each execution, while the deterministic ones produce always the same solution. MOM-aiNet is nondeterministic and several biclusters are given as the outcome at each run.

The biclusters returned by the algorithms can have different structures: (*i*) exclusive columns and/or rows, which consists of biclusters that cannot overlap in either columns or rows of the matrix; (*ii*) arbitrarily positioned and possibly overlapping biclusters, which is the case of MOM-aiNet; and (*iii*) overlapping biclusters with hierarchical structure.

Concerning the quality measure to be adopted, to calculate the coherence among the elements of a bicluster, it is used the mean squared residue, introduced by Cheng and Church [2]. This metric consists in the calculation of the additive coherence inside a bicluster by assuming that each row (or column) of the bicluster presents a profile identical to (or very similar to) the one exhibited by other rows (or columns), except for a constant bias. Therefore, finding a coherent bicluster is basically the same as finding a bicluster that minimizes the error between the calculated value and the real value of an element of the matrix. So the mean squared residue becomes $H(I, J)$:

$$H(I,J) = \frac{1}{|I||J|} \sum_{i,j} (a_{ij} - a_{Ij} - a_{iJ} + a_{IJ})^2, \tag{1}$$

where $|I|$ is the total number of rows of the bicluster, $|J|$ is the total number of columns of the bicluster, a_{ij} is the value in row i and column j, a_{Ij} is the mean value of column j, a_{iJ} is the mean value of row i, and a_{IJ} is the mean value considering all the elements of the bicluster.

Other important aspect of the biclusters is their *volume*, generally denoted in the literature by $|I| \times |J|$. In order to be functional and to allow a deeper analysis of the data, it is usually required that a bicluster presents a large volume (large number of rows AND columns).

Notice that minimizing the mean-squared residue and maximizing the volume are conflicting objectives, given that larger biclusters tend to present higher residues.

3 MOM-aiNet: Multi-Objective Multipopulation Artificial Immune Network

The aiNet algorithm was first proposed by de Castro & Von Zuben [3] to solve clustering problems, and it is based on two immune concepts: the Clonal Selection principle [17] and the Immune Network theory [18]. In aiNet, the population of candidate solutions corresponds to the *antibodies* of the system, while the data of the problem is associated with the antigens. In general terms, the population of antibodies is successively submitted to a cycle of *cloning*, *hypermutation* (with genetic variability proportional to the fitness of each individual – also known as the affinity of each antibody with the antigens of the problem) and *selection* [19]. Due to the immune network principles also followed by the algorithm, such

antibodies are capable of recognizing each other, so that if two individuals are similar, the worst one is eliminated from the population (in the phase known as *suppression*), and new randomly generated antibodies are inserted into the population to contribute with the diversity of solutions.

Such structure of the aiNet algorithm allows it to evolve multiple populations in parallel (associated with each antibody there is a population of mutated clones), and stimulates the convergence of such subpopulations to distinct promising regions of the search space (generally to the nearest local optimum), thus preserving diversity.

Given the two conflicting objectives of the biclustering problem and the usual necessity of extracting several biclusters from a single dataset, the multipopulation property of the aiNet algorithm will be exploited in this work, together with the concept of dominance, to create a novel algorithm capable of finding multiple biclusters while simultaneously optimizing both objectives and also obeying some pre-defined constraints.

The concept of dominance [20] is generally adopted to compare the quality of two solutions, of a given problem, when there is more than one objective being optimized. It is said that solution A dominates solution B (and so solution A is better than solution B) when A presents all the values of the objective functions better than or equal to the corresponding objective values of solution B, and there is at least one of the objectives for solution A that is strictly better than the equivalent for solution B. Therefore, it is possible to notice that three different situations can occur in a multi-objective problem: solution A dominates solution B (A is better than B); solution B dominates solution A (B is better than A); and A and B are mutually non-dominant. In a multi-objective optimization problem with conflicting objectives, the solution is, in fact, a set of non-dominated solutions, that correspond to the different trade-offs considering all the objectives. Current non-dominated solutions may be dominated by subsequent candidate solutions proposed along the execution of the algorithm. However, there is a set of solutions that will never be dominated by any feasible candidate solution, and they constitute a *front* in the objective space, which is known as the *Pareto Front* of the problem.

The aiNet family have already been successfully adapted to multi-objective optimization [21], but the approach presented in this paper will be significantly different from the one in [21]. omni-aiNet, besides additional attributes, is specialized in sampling the Pareto front uniformly and with high precision, while the algorithm to be presented in this section makes a rougher approximation of the Pareto front. In most multi-objective optimization algorithms, including omni-aiNet, a single set of non-dominated solutions is usually returned. But, in this paper, we propose a multi-objective and multipopulation immune-inspired approach, denoted *MOM-aiNet*, which returns several sets of non-dominated solutions (dominance is measured inside each set), each one potentially corresponding to biclusters extracting distinct correlations of rows and columns of the data matrix. As mentioned before, MOM-aiNet considers two objectives for optimization: the *residue* of the biclusters (which should be minimized) and their

volume (which should be maximized). The proposed algorithm is composed of the modules depicted in Alg. 1, which are going to be explained in the sequence.

Algorithm 1. The MOM-aiNet algorithm

```
generate_initial_population();
while stopping criterion is not met do
    for each population do
        clones = clone(smallest_bicluster & largest_bicluster);
        population_i = select_nondominated(clones + population_i);
    end for
    network_suppression();
    insert_new();
end while
```

The algorithm starts with the generation of n subpopulations of one bicluster each, generated by randomly choosing one row and one column of the dataset. If the dataset is sparse, the algorithm must choose only among the non-null values. Inside the main loop, for each subpopulation n_clones clones are generated, being half of the clones copied from the bicluster with the smallest volume in the subpopulation, and the other half from the bicluster with the highest volume.

Each clone then suffers a mutation process, which consists of one of three possible actions chosen randomly with the same probability: insert a row, insert a column, remove a row or column. Each action randomly selects one element to be inserted/removed. After the mutation step is performed on each clone, interpreted as a subpopulation of the algorithm, all the non-dominated biclusters of this subpopulation are selected to generate the new subpopulation, for the next iteration. If the number of non-dominated elements exceed n_clones, a crowding-distance-based [22] suppression is performed in order to maintain a small and locally diverse subpopulation.

The reason for the cloning process being performed only on the smallest and largest biclusters is the incremental nature of the mutation process, where only one row or one column can be inserted in/removed from each bicluster at a time. Therefore the biclusters are likely to suffer an incremental growth toward both ends (low and high volume), thus tending to be equally distributed on the non-dominated front.

Two constraints of the bicluster can be controlled on these steps. One is the residue value that can be limited to a specified value (δ), where every bicluster with a residue value higher than δ is said to be dominated by any other (unless there is only two or less biclusters on the population). And the other is the occupancy rate, for sparse matrices, that measures the proportion of non-null values in the bicluster. When an insertion action is chosen by the mutation process, the number of available rows/columns to insert is reduced to those that makes the bicluster occupancy rate no less than a threshold α.

After the cloning and mutation process, from time to time, a suppression operation is performed, so that the largest biclusters of each subpopulation are compared, based on the degree of overlapping. When a pair of biclusters have

a degree of overlapping higher than a given σ, the two subpopulations are combined and the non-dominance selection is performed, creating a single subpopulation. Only the largest bicluster of each population is taken for comparison because they tend to be more representative, and so two subpopulations will only be merged when they start to express the same correlations among rows and columns of the data matrix.

Finally, besides the suppression of similar subpopulations, it is also performed an insertion of new randomly generated subpopulations, in order to increase diversity and the exploration capability of the algorithm. This random insertion is performed in the same way as the initial subpopulations are generated, but with the difference that first it is chosen a pair of row and column that are not contained in any existing bicluster.

In the next subsection, a brief explanation of each one of the other three algorithms adopted in this work for comparison will be given.

3.1 Comments on BIC-aiNet, CC and FLOC

The BIC-aiNet and MOM-aiNet algorithms work in a very similar way, except that BIC-aiNet keeps just one bicluster per population (antibody) and there is no constraint on residue and occupancy. Also, the fitness function is a weighted sum of the two objectives being optimized (residue and volume).

The CC algorithm is a constructive heuristic that starts with a single bicluster, representing the whole dataset, and iteratively removes rows and columns of this bicluster until the residue is equal or less than δ. After that, it starts to insert rows and columns (that are not in the bicluster yet) sequentially, until the insertion of any other row or column increases the residue to a value above δ. After the first bicluster is constructed, the rows and columns already present in the bicluster are replaced by random values in the original dataset, and the whole process is restarted until a predefined amount of biclusters is created.

Finally, the FLOC algorithm tries to improve CC's mechanism by creating all the biclusters at the same time. The algorithm starts with the random generation of n biclusters with a predefined size, and then performs successive insertions or removals of each row and column of the dataset (according to the presence or absence of the row/column in the selected bicluster) in the bicluster that presents the highest reduction in the residue, when submitted to this modification.

4 Experimental Results

The MOM-aiNet algorithm was implemented in the C++ programming language and it was executed on an Athlon64 3500+ machine with 1GB of RAM. To evaluate its performance, first the algorithm was compared to FLOC, CC and BIC-aiNet algorithms on a dense dataset called Yeast microarray dataset [4], that contains $2,884$ genes under 17 experimental conditions. After that, MOM-aiNet was compared to BIC-aiNet on a sparse dataset called Movielens [5], that presents $80,000$ ratings of $1,682$ movies given by 943 users.

For the Yeast dataset, the results of FLOC and CC were taken from [6]. In this experiment, different values of the residue threshold (the δ parameter) were adopted for each immune-inspired algorithm (when compared to those in [6] for FLOC and CC), since each of these algorithms deals with this parameter differently. FLOC and CC continue the optimization process of the residue until they can not improve the solutions anymore, while the MOM-aiNet considers δ as a constraint and so generates the largest biclusters that present a residue lower than this threshold. Therefore, in order to make fair comparisons among the algorithms, this parameter was set as 185 for MOM-aiNet, which corresponds to a value a little lower than the average residue obtained by FLOC in [6]. The BIC-aiNet algorithm introduces δ in the definition of the fitness of the individuals, so that a value of 100 was empirically obtained.

The remaining parameters of MOM-aiNet and BIC-aiNet were also empirically determined as follows: for both algorithms it was adopted 1000 iterations, $n\_clones = 20$, and a maximum number of subpopulations/biclusters of 100; for the BIC-aiNet, the row importance weight was set to $w_r = 2$ and the column importance weight was set to $w_c = 3$ (further details about the parameters of BIC-aiNet can be found in [7]).

Table 1 presents the average results obtained by MOM-aiNet and BIC-aiNet, over 10 independent runs, together with the results from the other two algorithms (taken from [6]) on the Yeast dataset. Since MOM-aiNet is a multi-objective approach, each subpopulation generates up to n_clones biclusters so, in order to compare its results with the ones of the other algorithms, the average residue and volume of the largest bicluster of each subpopulation were taken in each independent run.

Table 1. Performance comparison among MOM-aiNet, BIC-aiNet, CC and FLOC algorithms for the Yeast dataset. The MOM-aiNet and BIC-aiNet results are shown in the format (average $\pm$ std. deviation), taken over 10 independent runs.

Algorithm	Avg. Residue	Avg. Volume
MOM-aiNet	**178.28 $\pm$ 5.24**	1831.80 $\pm$ 114.54
BIC-aiNet	194.65 $\pm$ 9.25	**2556.60 $\pm$ 188.92**
CC	204.29	1576.98
FLOC	187.543	1825.78

From Table 1, it can be seen the superior performance of the immune-inspired algorithms over CC and FLOC, since they can generate biclusters with lower residue values and/or higher volumes. It can also be seen that, differently from BIC-aiNet, the multi-objective approach was able to generate biclusters with residues close to the desired threshold and, at the same time, high volumes. This illustrates a disadvantage of the BIC-aiNet algorithm, more specifically the lack of control of the residue values of the generated biclusters.

Other advantage of the MOM-aiNet algorithm is that it also returns an array of different biclusters (the final individuals in each subpopulation), that correspond

to the compromise between volume and residue (non-dominated individuals) found in distinct regions of the data matrix (representing distinct correlations), which is very useful in the post-analysis process. In order to illustrate this set of biclusters returned by MOM-aiNet, Fig. 2 presents the residue and volume of the individuals in the final subpopulations obtained in one of the ten independent runs performed here, together with the final population generated by BIC-aiNet and the average results of FLOC and CC (taken from [6]).

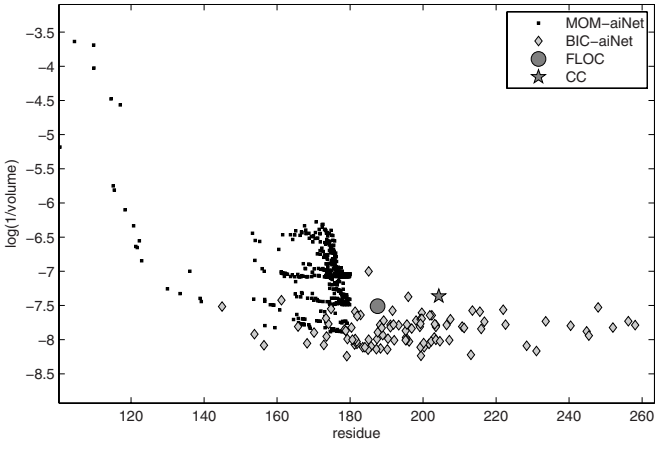


Fig. 2. 1/(Volume) (in logarithmic scale) and residue of the individuals in the final subpopulations of MOM-aiNet and of the final population of BIC-aiNet, for the Yeast problem. The results of FLOC and CC correspond to the average values of residue and volume of the final population of biclusters, as reported in [6].

As can be seen from Fig. 2 (and also from Fig. 3), the final individuals of different subpopulations returned by MOM-aiNet clearly present different quality (if only the two optimization criteria are considered) since several of them are dominated. However, it is important to notice that, although these individuals are dominated by others, they correspond to biclusters in different regions of the data matrix (otherwise, they would have been joined in a single subpopulation) and, consequently, are also of relevance in the post-analysis process.

It can also be observed in Fig. 2 that the individuals of the final population of the BIC-aiNet algorithm are concentrated in a region of higher volume when compared to the individuals returned by MOM-aiNet. However, most of these individuals are significantly above the required residue threshold ($\delta = 185$), what highlights the difference of both immune-inspired algorithms in the ability to control the residue values of the generated biclusters.

Next, on Table 2, the results obtained by MOM-aiNet and BIC-aiNet on the Movielens dataset (taken over 10 independent runs) are presented. This dataset is very sparse, which makes the generation of dense and coherent biclusters difficult. The parameters used for both algorithms were the same ones adopted

for the Yeast problem, except for the residue threshold and maximum number of subpopulations (biclusters), that were set to 2 and 300, respectively.

The results presented in Table 2 points out to the MOM-aiNet advantage of having a better control over the upper bound of the residue, what guarantees the generation of biclusters of higher volumes with residue values close to the desired one (threshold). The BIC-aiNet does not present an explicit control over this parameter, so it generates biclusters with a smaller residue, but also with a smaller volume.

Table 2. Performance comparison between MOM-aiNet and BIC-aiNet algorithms for the Movielens dataset. The results are shown in the format (average ± std. deviation), taken over 10 independent runs.

Algorithm	Avg. Residue	Avg. Volume
MOM-aiNet	1.26 ± 0.68	$\mathbf{203.88 \pm 23.51}$
BIC-aiNet	$\mathbf{0.43 \pm 0.03}$	83.39 ± 13.56

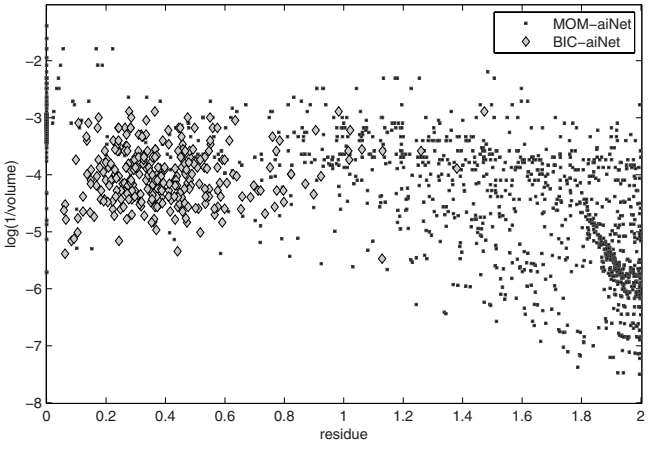


Fig. 3. Individuals in the final populations of MOM-aiNet, together with the final population of biclusters generated by BIC-aiNet for the Movielens dataset. The *volume* axis is in logarithmic scale.

Figure 3 presents the residue and volume of the individuals in the final subpopulations, together with the final population generated by BIC-aiNet, obtained in one of the ten independent runs performed here. As can be seen from Fig. 3, the individuals of the final population of the BIC-aiNet algorithm are concentrated on a smaller region of the objective space, and are clearly dominated by some individuals obtained by MOM-aiNet. It can also be seen that the region populated by the individuals returned by BIC-aiNet also presents individuals from MOM-aiNet (although in a smaller number), which illustrates that MOM-aiNet was also capable of covering the region explored by BIC-aiNet.

5 Final Remarks

In this paper, a novel multi-objective multipopulation artificial immune network for data biclustering, named MOM-aiNet, was proposed. The MOM-aiNet algorithm allows the generation of more than a single non-dominated front of solutions, each one corresponding to a different region of the original dataset.

The proposed algorithm was applied to two well-known datasets from the literature: a dense matrix called Yeast microarray data (for which MOM-aiNet was compared with BIC-aiNet, FLOC and CC algorithms); and a sparse dataset called Movielens (for which MOM-aiNet was compared with BIC-aiNet). The results have shown that the proposed algorithm was able to produce better results than the other algorithms on the two datasets, with the advantages of having a better control over the bicluster quality and also returning a broader set of non-dominated solutions. Besides that, MOM-aiNet together with BIC-aiNet also present the advantage of being easily divided into several parallel processes, which may be explored when dealing with larger datasets.

As future steps, more extensive sets of experiments will be performed with the MOM-aiNet algorithm and the biclusters generated by the proposed algorithm will be applied to collaborative filtering in the Web 3.0, which is a method to perform automated suggestions for a user, based on the opinion of other users with similar interests.

Acknowledgements

This research was sponsored by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, process number 20080129215000. The authors would also like to thank CNPq and CAPES for their additional financial support.

References

1. de França, F.O., Bezerra, G., Von Zuben, F.J.: New Perspectives for the Biclustering Problem. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Vancouver, Canada, pp. 753–760 (2006)
2. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Biology, pp. 93–103 (2000)
3. de Castro, L.N., Von Zuben, F.J.: aiNet: An Artificial Immune Network for Data Analysis. In: Abbass, H.A., Sarker, R.A., Newton, C.S. (eds.) Data Mining: A Heuristic Approach, pp. 231–259. Idea Group Publishing (2001)
4. Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., Davis, R.: A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell 2, 65–73 (1998)
5. GroupLens: MovieLens dataset. GroupLens, Department of Computer Science and Engineering, University of Minnesota, http://www.grouplens.org/node/73
6. Jiong, Y., Haixun, W., Wei, W., Yu, P.S.: Enhanced biclustering on expression data. In: Proc. of the Third IEEE Symposium on Bioinformatics and Bioengineering, pp. 321–327 (2003)

7. de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying Biclustering to Text Mining: An Immune-Inspired Approach. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 83–94. Springer, Heidelberg (2007)
8. de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying Biclustering to Perform Collaborative Filtering. In: Proc. of the 7th International Conference on Intelligent Systems Design and Applications, Rio de Janeiro, Brazil, pp. 421–426 (2007)
9. de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Evaluating the Performance of a Biclustering Algorithm Applied to Collaborative Filtering: A Comparative Analysis. In: Proc. of the 7th International Conference on Hybrid Intelligent Systems, Kaiserslautern, Germany, pp. 65–70 (2007)
10. Agrawal, R., Gehrke, J., Gunopulus, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. of the ACM/SIGMOD Int. Conference on Management of Data, pp. 94–105 (1998)
11. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc. of the 7th Int. Con. on Knowledge Discovery and Data Mining, pp. 269–274 (2001)
12. Feldman, R., Sanger, J.: The Text Mining Handbook. Cambridge University Press, Cambridge (2006)
13. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Statistical Association (JASA) 67(337), 123–129 (1972)
14. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Nearest-biclusters collaborative filtering with constant values. In: Advances in Web Mining and Web Usage Analysis, Philadelphia, USA. LNCS, vol. 4811, pp. 36–55. Springer, Heidelberg (2007)
15. Tang, C., Zhang, L., Zhang, I., Ramanathan, M.: Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In: Proc. of the 2nd IEEE Int. Symposium on Bioinformatics and Bioengineering, pp. 41–48 (2001)
16. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
17. Burnet, F.M.: Clonal selection and after. In: Bell, G.I., Perelson, A.S., Pimgley Jr., G.H. (eds.) Theoretical Immunology, pp. 63–85. Marcel Dekker Inc., New York (1978)
18. Jerne, N.K.: Towards a network theory of the immune system. Ann. Immunol., Inst. Pasteur 125C, 373–389 (1974)
19. de Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation 6(3), 239–251 (2002)
20. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2001)
21. Coelho, G.P., Von Zuben, F.J.: omni-aiNet: An immune-inspired approach for omni optimization. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 294–308. Springer, Heidelberg (2006)
22. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)

Viral System to Solve Optimization Problems: An Immune-Inspired Computational Intelligence Approach

Pablo Cortés, José M. García, Luis Onieva, Jesús Muñuzuri, and José Guadix

Ingeniería de Organización, Escuela Técnica Superior de Ingenieros, Seville University, Camino de los Descubrimientos s/n E-41092 Seville, Spain
{pca,jmgs,onieva,munuzuri,guadix}@esi.us.es

Abstract. This paper presents Viral System as a new immune-inspired computational intelligence approach to deal with optimization problems. The effectiveness of the approach is tested on the Steiner problem in networks a well known NP-Hard problem providing great quality solutions in the order of the best known approaches or even improving them.

1 Introduction

Artificial Immune Systems (AIS), introduced in [1], has a wide scope that covers from optimisation to classifiers or networks. AIS are a biological representation of the Natural Immune System (NIS). NIS protects the organism from dangerous extern agents such as viruses or bacteria. Antibodies try to protect the organism from such pathogens. Immune systems have a lot of peculiarities that make them very attractive for computational optimization. Examples are pattern recognition, auto-identification, diversity, autonomy, multilayered, cooperation, robustness, apprenticeship and memory, self-organization and integration among others. All these aspects make AIS attractive to manage optimization problems with constraints and objective functions. Some examples for combinatorial optimization problems are [2], [3] or [4].

Attending to these optimisation capabilities of AIS, we present Viral System (VS) that makes use of the same infection-antigenic response concept from immune systems, but from the perspective of the pathogen. In fact, real optimization problems are complex, especially those that are classified as NP-Hard. Several metaheuristics (as genetic algorithms, tabu search or simulated annealing among others) have successfully tried to deal with such problems. However, new research is being undertaken in order to find more successful methods to solve this kind of problems. Examples of that are Artificial Life algorithms, in particular predator prey type models, which are relatively closed to our VS, see [5] for an in-depth description of such models in a Multi-Agent System context.

The rest of the paper deals with the natural description of VS in section 2 where the natural immune characteristics of the algorithm are described, the computational aspects of the system in section 3, section 4 shows the results of VS when applying to the Steiner problem what is a well-known NP-Hard problem that was used as framework to test VS, and finally the main conclusions are detailed in section 5.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 83–94, 2008.

2 Virus Organisms and Virus Replication Mechanisms

Viruses are intracellular parasites shaped by nucleic acids, such as DNA or RNA, and proteins. The protein generates a capsule, called a capsid, where the nucleic acid is located. The capsid plus the nucleic acid shape the nucleus-capsid, defining the virus.

One of the main characteristics of viruses is the replication mechanism. The phage (a common type of virus) does follow lytic replication process. Right side of Fig. 1 depicts the biological evolution of the virus infection following the next steps:

1. The virus is adhered to the border of the bacterium. After that, the virus penetrates the border being injected inside this one, (a) and (b) in Fig. 1.
2. The infected cell stops the production of its proteins, beginning to produce the phage proteins. So, it starts to replicate copies of the virus nucleus-capsids, (c) and (d) in Fig. 1.
3. After replicating a number of nucleus-capsids, the bacterium border is broken, and new viruses are released, (e) in Fig. 1, which can infect near cells, (f) in Fig. 1.

The life cycle of the virus can be developed in more than one step. Some viruses are capable of lodging in cells giving rise to the lysogenic replication. This case is shown in the left side of Fig. 1. It follows:

1. The virus infects the host cell, being lodged in its genome, (g) and (h) in Fig. 1.
2. The virus remains hidden inside the cell during a while until it is activated by any cause, for example ultraviolet irradiation or X-rays, (i) in Fig. 1.
3. The replication of cells altered, with proteins from the virus, starts.

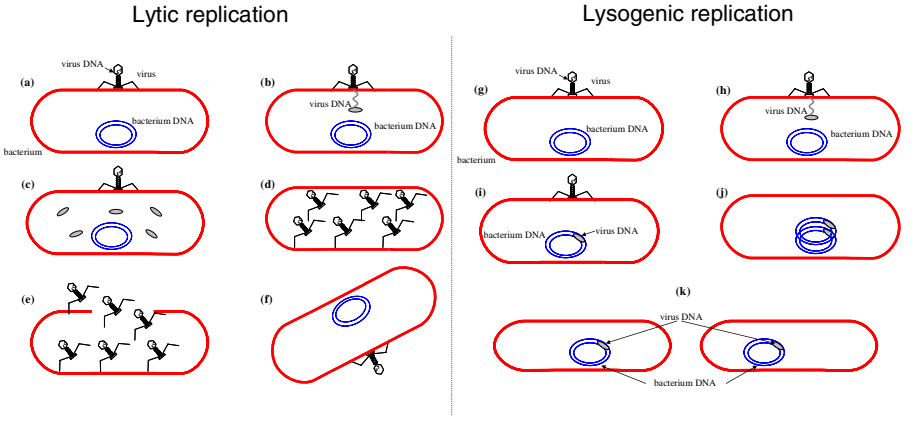


Fig. 1. Lytic (left) and lysogenic (right) replication of viruses

However, some viruses have the property of leading an antigenic response in the infected organism. In these situations an immune response is originated causing the creation of antibodies.

The main difference between VS and AIS is the final goal of the process. Meanwhile AIS are focused on the organism side, VS is focused on the virus side.

The optimum is got when the organism dies and the infection triumphs. So it follows other objective different from traditional AIS. Next section describes the process due to the interaction between the viruses and the organism.

3 Viral System Description

3.1 Viral System Components

VS are defined by three components: a set of viruses, an organism and an interaction between them: $VS = <Virus, Organism, Interaction>$.

The Virus component of the VS is a set consisting of single viruses: $Virus = \{Virus_1, Virus_2, \ldots, Virus_n\}$. And each virus is defined in four components: $Virus_i = <State_i, Input_i, Output_i, Process_i>$

Where each component means:

- $State_i$ characterizes the virus. It defines the cell infected by the virus. It is typically the mathematical encoding of the solution in computational terms, which we also call *genome*.
- A concrete virus, $Virus_i$, can produce the infection of a cell of the organism providing a host. Additionally, the evolution of the residence time of the virus inside the cell can be defined by the number of nucleus-capsids replicated for the lytic replication (NR) or the number of iterations for the lysogenic replication (IT). So, the three-tuple genome-NR-IT defines the $State_i$ for the $Virus_i$.
- $Input_i$ identifies the information that the virus can collect from the organism. This information is always collected in the proximity of the virus. $Input_i$ represents the input's interaction with the organism (organism's information → virus). It corresponds to the neighbourhood of the cell in computational terms.
- $Output_i$ identifies the actions that the virus can take. $Output_i$ represents the output's interaction with the organism (virus → organism). It corresponds to the selection mechanism of the type of virus replication in computational terms.
- $Process_i$ represents the autonomous behaviour of the virus, changing the $State_i$. It corresponds to the replication operator process in computational terms.

The *Organism* component of the VS is defined by two components:

$$Organism = < State_o, Process_o>$$

Where each component means:

- $State_o$ characterizes the organism state in each instant. It consists of the clinical picture and the lowest healthy cell (the best solution found of the optimization problem). The set of feasible solutions in a specific space $\Re^n$ is given by the problem constraints (1).

$$K = \{x : g_i(x) \leq 0,\ \forall i = 1, \cdots, n\} \tag{1}$$

Each feasible solution of problem (1), $x \in K$, has been called a cell. The *genome* is the mathematical encoding of each cell or feasible solution. When a virus infects a cell, this cell enters the population of infected cells. The total amount of infected cells constitutes the infected part of K for each time instant, and it is named "*clinical picture*". It contains the overall information of the infection needed by the algorithm

in each instant, *t*. Thus, the clinical picture consists of every three-tuple genome-NR-IT defining the $State_i$ of each $Virus_i$.

In the same way, the overall clinical picture plus the lowest value of $f(x)$ defines the *Organism* $State_o$. Fig. 2 depicts the *State* concept for the organism and the viruses.

- $Process_o$ represents the autonomous behaviour of the organism that tries to protect itself from the infection threat, consisting of antigen liberation. Medically, an antigen is any substance that elicits an immune response. The antigens generate an immune response by means of antibodies trying to fight the virus infection. The computational mission of the antigens is to liberate space in the population of infected cells (clinical picture), trying to maintain free record memory in the clinical picture to incorporate new infected cells (new feasible solutions). Thus, due to the antigens' activity, infected cells (in the clinical picture) can be recovered (removed) and cells in the organism that could be infected are not infected due to this antigenic substance.

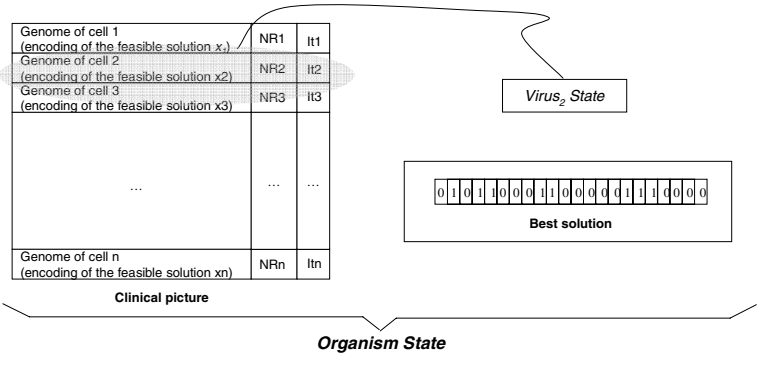


Fig. 2. Organism and virus state

3.2 Viral System Interaction

The *Interaction* component of the VS is conditioned by the *Input* and *Output* actions that lead to a *Process* of every virus and the corresponding *Organism* response. A $Virus_i$ process implies a resulting change in the organism, and the same applies for an *Organism*'s process. The interaction is the union of both actions.

3.2.1 Virus Input Sensor: Neighborhood Identification

The input sensor of each virus, $Input_i$, collects information from the organism. The sensors map the genome of the cell and detect the set of cells close to the infected one. This set is named the neighbourhood of the feasible solution x, $V(x)$. The neighbourhood depends on the shape of the constraints of the problem, $g_i(x)$.

3.2.2 Virus Output Ejector: Replication Type Selection

The ejector, $Output_i$, selects the type of evolution of the virus. We consider one step lytic replication (probability p_{lt}); and two steps lysogenic replication (probability p_{lg}). See Fig. 1. Where $p_{lt} + p_{lg} = 1$.

3.2.3 Process: Lytic Replication

3.2.3.1 Virus process. The lytic replication starts only after a specific number of nucleus-capsids have been replicated. So, each time instant (iteration t) a number of virus replications (NR) takes place. The number of replications per iteration is calculated as function of a binomial variable, Z, adding its value to the total NR.

After a specific number of nucleus-capsids has been replicated inside the cell (LNR), the bacterium border is broken, liberating the lodged viruses. All these viruses are active and prepared to infect new cells. The value of LNR depends on the cell's health conditions. So a healthy cell (with high value of $f(x)$) will have low probability of getting infected, and therefore the value of LNR will be higher. In the opposite it will have a lower value of LNR. Following equation (3) shows the calculation procedure for LNR in a cell x:

$$\text{LNR}_{\text{cell-}x} = \text{LNR}^{0} \cdot \left(\frac{f(x) - f(\hat{x})}{f(\hat{x})} \right) \tag{2}$$

where $\hat{x}$ is the cell that produces the best known result of the problem (in terms of $f(x)$) and x is the infected cell being analysed.

LNR^{0} is the initial value for LNR

The number of nucleus-capsids replicated each iteration can be approximated by a Binomial distribution given by the maximum level of nucleus-capsids replicated, LNR, and the single probability of one replication, p_r,: Z = Bin (LNR , p_r).

Once the distribution has been stated, we can calculate the probability of replicating exactly z nucleus-capsids, P(Z=z), as well as the average, E(Z), and variance, Var(Z), equations (3-5).

$$\text{P}(\text{Z} = z) = \binom{\text{LNR}}{z} p_r^{z} \cdot (1 - p_r)^{\text{LNR}-z} \tag{3}$$

$$\text{E}(\text{Z}) = p_r \cdot \text{LNR} \tag{4}$$

$$\text{Var}(\text{Z}) = p_r \cdot (1 - p_r) \cdot \text{LNR} \tag{5}$$

Once the number of nucleus-capsids surpasses the limit given by LNR, the border of the cell is broken and the viruses are liberated. For this case, one single cell is selected to be infected. In order to do so, the neighbourhood is evaluated and one of the less healthy cells is selected, configuring the new host to expand the infection.

3.2.3.2 Organism process. In this case, the virus selects a cell with a low value of f(x) in the neighbourhood. However, the virus will not be able to infect those cells that have developed antigens.

Higher values of $f(x)$ imply healthy cells and therefore cells that have a higher probability of developing antigenic responses. On the contrary, cells with low value of $f(x)$ imply unhealthy cells with lower probability of developing antigenic responses.

In order to represent such phenomenon, we use a hypergeometric function. The cell with an inverse objective function evaluation, $1/f(x)$, in ranking position-i, has a probability of generating antibodies, $p_{an}(x)$, that is given by $q(1\text{-}q)^i$, being q the probability of generating antibodies for the worst individual. Finally, a residual probability remains, which is added to the worst individual.

Then, if the probability of generating antibodies for the case of cell x is $p_{an}(x)$, $A(x)$ is defined as a Bernoulli random variable: $A(x) = \text{Ber}(p_{an}(x))$.

If cell x generates antibodies, the cell is not infected and it is therefore not included in the new clinical picture. For recording this clinical picture we use the original cell (that was infected by the virus and that reached the LNR limit) and we initiate a lysogenic cycle for that cell.

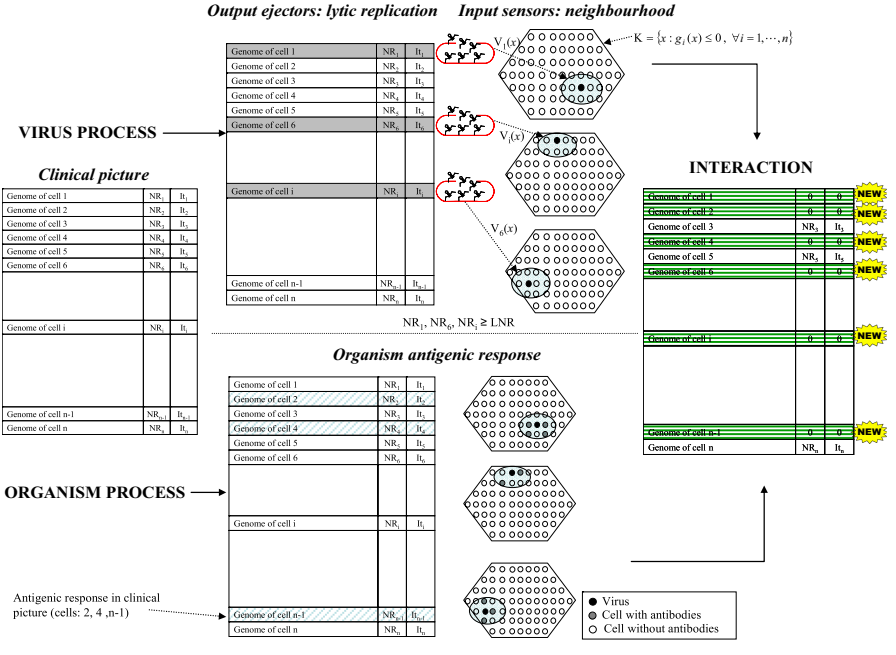


Fig. 3. Virus process evolution during the lytic cycle

Fig. 3 defines the algorithm evolution for the infection. The initial state is on the left-hand side: the virus process starts with viruses breaking the border and starting the infection of new cells in their neighbourhoods. Each virus selects the most promising cell, which is the least healthy cell. The Organism process is characterized by the probability of antigenic response in the least healthy cell. Those cells developing antibodies are not infected. Finally, the interaction (right hand side of the figure) defines the new clinical picture, with new infected cells lodging viruses. The cells generating antibodies follow a new lysogenic replication.

3.2.4 Process: Lysogenic Replication

3.2.4.1 Virus Process. During the lysogenic cycle the virus remains hidden inside the cell until an external cause activates the virus. We consider that the activation of the lysogenic replication can happen after a limit of iterations has passed (LIT). As for the calculation of LNR, the value of LIT depends on the cell's health conditions, so a healthy cell (high value of f(x)) will have a low probability of getting infected, id est. the value of LIT will be higher. On the contrary, it will have a lower value of LIT. Equation (6) shows the calculus procedure for LIT in a cell x:

$$\mathrm{LIT}_{\text{cell-}x} = \mathrm{LIT}^0 \cdot \left(\frac{f(x) - f(\bar{x})}{f(\bar{x})} \right) \tag{6}$$

Where LIT^0 is the initial value for LIT

Once the virus has been activated, it produces alterations in the cell's genome. It is equivalent to a genome mutation process in the mathematical programming encoding of the feasible solution.

3.2.4.2 Organism Process. The lysogenic interaction is described as the substitution of the new genome-modified cell by the old one. It is quite similar to a mutation process in several types of evolutionary algorithms.

3.3 End of the Biological Process

The VS ending is achieved in two ways: the organism beats the virus implying the host recovery, or the virus beats the defence capabilities of the organism and the host death takes place.

Computationally, the ***death of the organism*** can be reached when the difference between the best found solution and a known lower bound is smaller than a stated gap. There exist certain lower bounds known for several NP-problems. Nevertheless, a lower bound could always be calculated by means of the linear or Lagrangian relaxation for problems with a linear objective function and linear constraints. In case of knowing the optimum of the problem, the gap can be set equal to zero. This is a common case when dealing with trial problem collections.

When the difference between the lower bound and the best found solution is below a *gap*, we consider that the organism has collapsed (7), and the VS infection ends.

$$gap = \frac{|f(x^*) - \mathrm{LB}|}{\mathrm{LB}} \tag{7}$$

Other possible end appears after reaching a maximum number of iterations (Nmax), we consider that the viral infection cannot evolve further and the ***virus is isolated***.

When this criterion is used together with the previous one, the situation denotes that the gap is not reached, and the virus does not create a serious infection in the organism. Under this condition, the organism would have survived the virus infection.

3.4 Viral System Pseudocode Algorithm

Table 1 details the main functions of the VS pseudocode.

Table 1. VS pseudocode

```
Initiate clinical_picture
Select infection type
Initiate iterations
Do {
           if case_infect = massive
{          antigen(clinical_picture)
}
           replicat_type(clinical_picture)=
output(clinical_picture)
           Do {
           if (replicat_type(cell) = lytic)
             { replicate(cell)
               NR = NR + replicate(cell)
               if (NR ≥ LNR)
               { neighbourhood(cell) = input_lytic(cell)
                 cell_infected=
process_virus(cell,case_infect)
                 update_clinical_picture(cell_infected)
               }
             }
             else
             { iter = iter + 1
if (iter ≥ LIT)
mutate_genome(cell) = input_lysogenic(cell)
                 update_clinical_picture(mutate_genome)
             } while (clinical_picture)
                                    } while{ gap () OR Nmax}
```

4 Computational Results: The Steiner Problem in Networks

We used the Steiner problem (SP), a well-known NP-Hard problem to test VS. SP is stated as follows: given a non-directed graph $G = (N,A)$ with $|N|$ nodes and $|A|$ arcs with costs c_{ij} $\forall(i,j)\in A$; and a subset $T\subseteq N$ with $|T|$ nodes called terminals or targets, with the rest of the nodes in N called Steiner nodes, the goal is to find a network $G_T \subseteq G$ joining all the terminal nodes in T at minimum cost. This network can include some of the Steiner nodes but does not have to include all the Steiner nodes.

4.1 VS Characterization for the Steiner Problem

The ***Organism state*** is depicted by the clinical picture representing the infected part of the SP hull, K. A coverage formulation for the SP is shown in equation (8), [6].

$$K : X\left(\delta(W)\right) \geq 1\ ,\ \forall W \subset N\ ,\ W \cap T \neq \varnothing\ ,\ \left((N \backslash W) \cap T \neq \varnothing\right)$$
$$0 \leq x_{ij} \leq 1\ ,\ \forall (i,j) \in A\ ;\ x \text{ integer} \tag{8}$$

Where $\delta(X)$ denotes the cut induced by $X \subseteq N$, that is, the set of arcs with one node in W and one in its complement. It is easy to see that there is a one-to-one correspondence between Steiner trees in $G = (N,A)$ and $\{0,1\}$ vectors satisfying K.

We represent the ***genome*** of the cells by a bit string of size equal to $|N|$ in which each bit position i corresponds to the node i in the graph. A 1 means that the node i is connected, while the bit is set to 0 otherwise. As all the terminals must be in the Steiner tree, it is sufficient to use a bit string of size $|N\text{-}T|$ including only the Steiner nodes belonging to the Steiner tree. So, the Steiner tree can be constructed by a minimum spanning tree that contains all the terminal nodes (set T), the subset of Steiner nodes in the bit string fixed to 1 and, perhaps, some artificial arcs if the set is disconnected. We made use of the graph construction mechanisms described in [7].

Once we have stated the cell genome we can define the ***Virus state***. The three-tuple formed by the genome of each cell infected plus the number of replicated nucleus-capsids (in the case of lytic replication) or the number of generations (in the case of lysogenic replication) defines the virus state. The entire infected cell population, which is the clinical picture, and the best solution complete the *Organism* and therefore the *Virus* state.

The ***Output ejectors*** of the *Virus* component of the VS are clearly defined by the type of replication. On the contrary, the ***Input sensors*** must be carefully stated. In fact, a key decision is to state an adequate cell neighbourhood for the virus in the lytic replication process and a genome alteration process for the lysogenic replication.

In case of the Steiner problem, the lysogenic replication is characterized as a genome alteration by flipping a bit in the string. The lytic replication for a feasible solution $x \in K$, maps the neighbourhood consisting of the set of bit strings that can be obtained by the removal or the addition of a single Steiner node from/to the current cell encoding. In order to be efficient, the new MSTs must be found by manipulating a rooted tree data structure carefully, [7].

Finally, the *Virus* and *Organism* components are completed by the specification of the *Process*. The ***Organism Process*** consists of the antigenic response and it is mainly determined by the determination of the parameter p_{an}. The ***Virus Process*** consists of the determination of the type of replication that is conditioned by the parameters p_{lt} and p_{lg}. Additionally, the *Virus Process* depends on the parameters of replication, p_r, infection, p_i, and the limits LNR^0 and LIT^0. Due to the special encoding for the Steiner problem solutions the neighbourhood size is constant and equal to the number of Steiner nodes. It must be noted that the neighbourhood is set by changing the value of each bit from 0 to 1 and vice versa.

The ***Interaction*** takes place after the selection of the *Virus Process*. It depends on the random evolution of the viral infection and the antigenic capacity of response.

4.2 Results

We used the OR-Library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html) for the Steiner problem: series C, D and E, each one of them including 20 problems. Steiner

series C consists of trials with 500 nodes, arcs varying from 625 to 12,500, and terminals from 5 to 250; series D consists of problems with 1,000 nodes, arcs varying from 1,250 to 25,000, and terminals from 5 to 500; and finally series E includes trials of 2,500 nodes, arcs varying from 3,125 to 62,500, and terminals from 5 to 1,250.

Table 2. Results: a comparison among Genetic Algorithm, Tabu Search and Viral System

Problem	GA-E	F-Tabu	VS
C1	0.00%	0.00%	0.00%
C2	1.67%	0.00%	0.00%
C3	0.13%	0.00%	0.00%
C4	0.11%	0.00%	0.00%
C5	0.00%	0.00%	0.00%
C6	0.73%	0.00%	0.00%
C7	1.76%	0.00%	0.00%
C8	0.63%	0.00%	0.00%
C9	1.05%	0.14%	0.00%
C10	0.26%	0.00%	0.00%
C11	1.88%	0.00%	0.00%
C12	1.30%	0.00%	0.00%
C13	1.01%	0.00%	0.00%
C14	0.87%	0.31%	0.00%
C15	0.25%	0.00%	0.00%
C16	0.00%	0.00%	0.00%
C17	0.00%	0.00%	0.00%
C18	0.71%	0.00%	0.00%
C19	0.41%	0.00%	0.00%
C20	0.00%	0.00%	0.00%
Total	**0.64%**	**0.02%**	**0.00%**

Problem	GA-E	F-Tabu	VS
D1	0.57%	0.00%	0.00%
D2	0.00%	0.00%	0.00%
D3	0.92%	0.06%	0.00%
D4	0.52%	0.00%	0.00%
D5	0.12%	0.00%	0.00%
D6	0.00%	0.00%	0.00%
D7	1.94%	0.00%	0.00%
D8	1.55%	0.37%	0.47%
D9	0.50%	0.21%	0.69%
D10	0.13%	0.00%	0.00%
D11	2.07%	0.00%	0.00%
D12	0.00%	0.00%	0.00%
D13	0.56%	0.00%	0.00%
D14	0.30%	0.15%	0.15%
D15	0.16%	0.00%	0.00%
D16	0.00%	0.00%	0.00%
D17	0.00%	0.00%	0.00%
D18	1.26%	0.90%	0.90%
D19	1.03%	0.32%	0.65%
D20	0.15%	0.00%	0.37%
Total	**0.59%**	**0.10%**	**0.16%**

Problem	GA-E	F-Tabu	VS
E1	0.00%	0.00%	0.00%
E2	0.93%	0.00%	0.00%
E3	0.00%	0.32%	0.24%
E4	0.02%	0.00%	0.00%
E5	0.00%	0.00%	0.00%
E6	0.00%	0.00%	0.00%
E7	0.00%	0.00%	0.00%
E8	0.23%	0.42%	1.14%
E9	0.19%	0.14%	0.47%
E10	0.00%	0.04%	0.14%
E11	0.00%	0.00%	0.00%
E12	1.49%	1.49%	0.00%
E13	0.70%	0.63%	1.33%
E14	0.23%	0.23%	0.64%
E15	0.00%	0.11%	0.00%
E16	0.00%	0.00%	0.00%
E17	0.00%	0.00%	0.00%
E18	3.37%	1.60%	2.66%
E19	1.26%	1.19%	1.18%
E20	0.00%	0.00%	0.15%
Total	**0.42%**	**0.31%**	**0.40%**

Table 2 shows the results (in error percentage with respect to the optimum) for the Stein-C, Stein-D and Stein-E problems and the comparison with the best Tabu Search approach from [7] (the F-Tabu method), which is the best approach for the Steiner problem in terms of solution quality. Additionally we have selected the best biologically inspired method to solve the Steiner problem. It is the case of the Genetic Algorithm approach due to [8], (GA in the table).

Each problem was four times solved and results in Table 2 show the best value reached in all the considered replications for VS, GA or TS. The standard deviation of the solutions provided by the methods was less than 5% as average value what is a feasible value for a design problem (SP).

According to the results of Table 2 (a total of 60 problems), VS was the best approach in 48 times and outperformed the GA-E (22 times) approach. F-Tabu showed better performance, being the best approach 51 times. However, VS provided a better solution for the C9, C14, D3, E3, E12, E15 and E19 problems. VS provided very valuable results taking into account that F-Tabu was processed after selecting the 100 best different trees found by the MPH algorithm, after executing the P-Tabu approach as an initial search and reprocessing it into the final Full Tabu Steiner (F-Tabu). So the quality of the F-Tabu results is very high but it is also very much conditioned by the very good seed that is provided. On the contrary, we applied VS directly to the graph without pre-processing it with any special previous heuristic as MPH or previous metaheuristics as P-Tabu. Nevertheless, the initial clinical picture of VS was wholly random-generated. We did not use a good seed provided by a good heuristic because we were interested in observing the quality of the VS evolution to the final solution, more than on outperforming previous heuristics. However, we realized that without searching for a good seed we were obtaining results equivalent (in quality terms) to the best Steiner approach: the F-Tabu algorithm.

With respect to the time consumption, we have to say that time values among methods cannot be directly compared because tests were run in a different computers. However, we can estimate the order of time consumption by the algorithm's complexity, given in (9).

$$time \sim O\left(ITER \cdot NumSteiners \cdot Ngraph^2\right) \tag{9}$$

Where *ITER* is the maximum number of iterations, *NumSteiners* the number of Steiner nodes in the graph and *Ngraph* the total number of nodes in the graph.

The solutions were attained using the parameters of Table 3. We found that VS efficiency was non-dependent on the probability of generating a great or low number of nucleus-capsids (parameter P_Z), so its performance showed non-dependency from this parameter in the SP case. The rest of parameters depended on the percentage of terminals mainly. So, two set of parameters were considered. We executed four times the VS with the first set, and four additional times for the second set of parameters.

Table 3. Parameters selection for VS

	% Terminals < 15%		**% Terminals ∈ [15%,30%]**		**% Terminals 30%**	
Parameters	1st set	2nd set	1st set	2nd set	1st set	2nd set
ITER	50,000	10,000	50,000	50,000	10,000	50,000
POB	100	50	100	50	50	50
PLITI	0.7	0.7	0.7	0.7	0.7	0.7
LNR	15	15	15	15	10	10
LIT	10	10	20	10	10	10
Pz	0.5	0.5	0.5	0.5	0.5	0.5

5 Conclusions

We have presented a new approach to optimize combinatorial problems called Viral System which is inspired in a natural immune system. The main difference between VS and traditional AIS is the final goal of the process. Meanwhile AIS are focused on the organism side, VS is focused on the virus side. The optimum is got when the organism dies and the infection triumphs. So it follows other objective different from traditional AIS. The method was tested with an extremely difficult combinatorial problem as the Steiner problem is. It is a well-known NP-Hard problem. In fact, most of the network problems are proved to be NP-Hard by reduction to the SP.

VS was applied to a large set of trials and was compared with the best approaches to solve the SP. VS clearly improved the results from the Genetic Algorithms (a bio-inspired evolutionary methodology close to our proposal) and also outperformed several times the Tabu Search approach (the best known metaheuristic for the SP).

Our future research is focused on applying VS to other well-known NP-Hard problems that arises in contexts different from networks in order to test its efficiency.

References

1. Farmer, J.D., Packard, N., Perelson, A.: The immune system, adaptation and machine learning. Physica D 22, 187–204 (1986)
2. Cutello, V., Nicosia, G., Pavone, M.: An Immune Algorithm with Stochastic Aging and Kullback Entropy for the Chromatic Number Problem. Journal of Combinatorial Optimization 14(1), 9–33 (2007)
3. Cutello, V., Nicosia, G., Pavone, M., Timmis, J.: An Immune Algorithm for Protein Structure Prediction on Lattice Models. IEEE Transaction on Evolutionary Computation 11(1), 101–117 (2007)
4. Cutello, V., Narzisi, G., Nicosia, G., Pavone, M.: Clonal Selection Algorithms: A Comparative Case Study using Effective Mutation Potentials, optIA versus CLONALG. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 13–28. Springer, Heidelberg (2005)
5. Van Dyke Parunak, H.: Go to the ant: Engineering principles from natural multi-agent systems. Annals of Operations Research 75, 69–101 (1997)
6. Koch, T., Martin, A.: Solving Steiner tree problems in graphs to optimality. Networks 32(3), 207–232 (1998)
7. Gendreau, M., Larochelle, J.-F., Sansò, B.: A tabu search heuristic for the Steiner tree problem. Networks 34(2), 162–172 (1999)
8. Esbensen, H.: Computing near-optimal solutions to the Steiner problem in a graph using genetic algorithm. Networks 26, 173–185 (1995)

Computing the State of Specknets: Further Analysis of an Innate Immune-Inspired Model

Despina Davoudani, Emma Hart, and Ben Paechter

Napier University, Scotland, UK
{d.davoudani,e.hart,b.paechter}@napier.ac.uk

Abstract. Specknets consist of hundreds of miniature devices, which are each capable of processing data and communicating wirelessly across short distances. Such networks, with their great complexity, pose considerable challenges for engineers due to the unreliability and scarce resources of individual devices. Their limitations make it difficult to apply traditional engineering approaches. In this paper, we describe a model inspired by the dendritic cells of the innate immune system; often overlooked in artificial immune systems, dendritic cells possess a unique ability to scout the body environment and then present an integrated picture of the internal state of the body to the adaptive system. We adopt a model, inspired by this approach, to sense the state of a Specknet and provide experimental results to show that useful information can be gathered from the Specknet in order to determine local states. Experiments are conducted using realistic random topologies in a simulation environment, in a scenario which models sensing temperature changes.

1 Introduction

Specks are autonomous, minute semi-conductor grains of around $5x5mm^2$ which possess the capability to sense, process, and transmit data via wireless sensor networking. Platforms consisting of thousands of such specks, termed *Specknets*, offer the potential of truly ubiquitous computing in which collaboration between specks results in programmable computational networks [2]. A Specknet fully embraces the idea of an autonomous system with programmable constituent parts, where the parts themselves are sustaining the system by showing 'self' properties such as self-organisation, self-sufficiency and self-adaptation. The network lacks powerful central processing units and relies on each programmable speck to process and act on information, in collaboration with its neighbours over short communication ranges of the order of tens of centimetres. These requirements pose considerable challenges to application developers, on top of the non-trivial challenges posed by the development of the specks themselves.

Biology, and in particular the immune system, provides a rich and obvious source of inspiration for working with such systems, given the similar requirements for meaningful behaviours to emerge from interactions of substantial numbers of individually weak entities. This has already been recognised in the field of wireless sensor networks (WSNs) [7] and was first proposed as a viable metaphor

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 95–106, 2008.

for programming Specknets in [6]. In this paper, we present clarification and further results from a model first described in [5], which takes inspiration from the *innate* immune system. Despite having received relatively scant attention from the community of artificial immune systems (AISs) until recently, the innate immune system exhibits many appealing features from a system perspective. Once thought to simply provide an indiscriminate, rapid defence until an adaptive response kicked in, it is now clear that the innate system is actually responsible for activating the adaptive system; this occurs as a result of scouting the body environment and presenting an integrated picture of the internal state of the body to the adaptive system, which is then triggered to react, or suppressed, according to the state information. This metaphor is ripe from exploration in a *system-based* application such as a network exposed to both external and internal signals, as in the body.

This paper presents extensions to the model described in [5] and provides further initial results obtained in random Specknet topologies, using a simple scenario in which a Specknet equipped with temperature sensors monitors externally applied fluctuations. In particular, Sect. 2 discusses related work; Sect. 3 describes the immunological theory that supports the model for a Specknet, which is presented in Sect. 4; Sect. 5 describes the simulation setup used to obtain the results presented in Sect. 6; finally the conclusions are discussed Sect. 7.

2 Related Work

Although the literature contains a wealth of work relating to biologically inspired approaches to WSNs and to immune-inspired algorithms in many diverse domains, there is little which is directly relevant to the use of immune systems in the class of WSNs typified by Specknet. We briefly mention some work which has some similarities, although as far as we can ascertain, our work is novel in the use of innate immune-inspired mechanisms to WSNs.

The SASHA architecture, proposed by [4], presents a self-healing hybrid sensor network architecture which is inspired by the natural immune system. This work is motivated by the same ideas as our work in taking a holistic approach to the immune system; the architecture is implemented on a classic sensor network and is directed towards achieving fault tolerance and adaptability to pathogens. The model incorporates many features of the immune system, but includes the use of high-powered database components and base stations which are not included in the vision of a Specknet, and hence reduces the applicability of the approach.

Atakan et al. [3] employ a method inspired by the behaviour of B cells in the adaptive immune system to distributed node and rate selection in a WSN. The aim is to select appropriate sensor nodes and regulate reporting frequencies to meet the event estimation distortion constraint at sink nodes, with the minimum number of sensor nodes. Essentially, the problem of reducing the amount of redundant information transmitted through the network is treated as an optimisation problem, with competitive selection acting on nodes in order to determine which nodes are best placed to transmit.

Finally, although not concerned with sensor networks, at a high-level our work has much in common with the dendritic cell algorithm (DCA) proposed by Greensmith *et al.* [8]. However, the implementation details differ at a low-level. This work was the first to exploit one of the essential properties of the innate immune system, that of its ability to perform sensor fusion of data and signals in order to determine system state. The DCA was proposed in the context of performing anomaly-detection in computer-security; although we do not utilise their algorithm itself, our approach captures the same essential property; that of gathering context dependent information over time in order to determine the state of a system. The DCA's main function is in determining the context of collected data; currently our approach focuses more directly on the gathering process of the data itself in a difficult, distributed environment. However, in future, the DCA may be employed more directly.

3 The Innate Immune System

As briefly described in Sect. 1, the innate immune system is a key component in a natural immune system. The innate system consists of a number of players, which collectively contribute to its overall functionality. In this section, we provide a brief, high-level overview of the role of one of those players in this system, the *dendritic cell*. The description necessarily omits much of the biological detail; the aim is simply to provide sufficient understanding of the processes that occur in the natural immune system to motivate the inspiration for our current work in WSNs.

The dendritic cell is often referred to as the 'sentinel' of the immune system [10], playing a unique role in sampling the body's tissues and reporting back on the state of them to the next line of defence, the adaptive immune system. Dendritic cells reside in the epithelial tissues of the body (e.g. the skin), sampling the tissue in their vicinity. Essentially, they soak up molecular debris (e.g. bacteria or other pathogenic material) and, additionally, sense molecular signals present in the tissue. The signals may derive from 'safe' or 'normal' events (e.g regular, pre-programmed cell death) or from potentially dangerous events, where cell death occurs due to stress or attack. These signals may be exogenous and/or endogenous. Whatever their source, collection of 'sufficient' signal and antigen triggers immature dendritic cells to mature. At this point, they travel back to the nearest lymph node through a complex system of lymphatic vessels.

The lymph nodes function as dating agencies where the different immune cells of the body congregate. In particular, the dendritic cells that reach the lymph node carry a snapshot of the current state of the tissues back. The snapshot contains two important pieces of information: antigen, i.e. (potentially) unsafe material, and also signals representing the context under which the material was collected. This snapshot is viewed by the reactive immune cells, in particular T cells, and a process of communication and collaboration between cells ensues. This process ultimately results in activation, or tolerance, of the immune system, depending on the content and context of the information presented. From the perspective of our research in Specknets, we identify two distinct roles of the innate immune system; firstly,

the physical role of traversing the immune system and gathering information, and secondly, the process of presenting that information to the adaptive system in the correct context. In this paper, we concentrate on the former role. In future work, we turn our attention to extracting the context of the gathered information and actuating the system to react accordingly.

4 An Immune-Inspired Model for a Specknet

The innate immune system processes, described in Sect. 3, offer three major sources of inspiration for the current model.

- Dendritic cells circulate through body tissues, sampling exogenous and endogenous signals.
- The dendritic cells return to the lymph nodes, via a process of chemotaxis, when they become mature, where they deliver a snapshot of the current environment.
- The lymph nodes in the body are distributed; the large lymph nodes are strategically located to areas of the body that are closer to sources of input from the environment.

From this, we derive a model consisting of *specks* and *scouting messages* (SMs). We currently distinguish between two different types of specks:

- *Tissue* specks correlate to tissues in the body, and contain sensors for monitoring the external environment (e.g. light, pressure, temperature etc.). They can also provide internal signals, for example relating to their own state (i.e. battery power). These specks constitute the majority of specks in any given environment.
- *Integration* specks correspond to lymph nodes. These specks receive information from dendritic cells, process it, and determine an appropriate response. These specks may have greater processing power than tissue specks.

A typical environment will contain a high ratio of tissue specks over integration specks. Although in the body lymph nodes are strategically placed, this is not feasible in a typical speck deployment, where thousands of specks may be sprayed at random into an environment. Therefore, we model random placements of integration specks. Dendritic cells are mapped to SMs. Messages originate at integration specks and traverse the tissue specks, where they collect information from each speck visited. Eventually, they return to the integration specks, where the information collected is processed. The implementation of these processes are discussed below.

4.1 Traversal of Tissue Specks

Messages originate from the integration speck and follow a random walk through the tissue. The walk is achieved by using one-hop neighbourhoods. Each speck maintains a list of IDs that lie within radio range, from where it randomly selects

the next speck ID to be visited by the SM. If the destination hop is a tissue speck, the SM collects one more sample, otherwise the integration speck simply relays the message to another random neighbour. The technical details regarding the manner in which a speck obtains and maintains its list of neighbours are described in [5].

Aggregating information from the network via SMs is configurable with respect to the type of information needed. With every SM it releases, the integration speck can specify the type of *local (L)* and *group (G)* samples to be gathered. A local sample is the information a SM requests from each tissue speck visited, and may for example take the form of an average value of sensor readings over a specified window size. The collection of local samples is returned to the integration speck, in a form known as group sample, by applying a function to the set of local samples (e.g. an average, minimum or maximum). The parameters used for processing sensor readings from the tissues and the functions applied to local sensor values are application dependent. With respect to group samples, the processing of local samples can be performed as the SM passes through the network, reducing the required data rate. However, in other cases this reduction may not be possible, for example when more complex functions need to be applied.

The life cycle of a SM is shown in Fig. 1. Scouting messages that sample the network are in a naive state, and can either become mature, as a result of collecting *interesting* information, [1] or simply expire; in either case they return to the lymph to present their information. Information gathered by expiring SMs is of relevance to the integration speck, which can estimate context based on the proportion of expired to mature messages returning, and also by aggregating information contained in the expired messages.

4.2 Chemotaxis Back to Integration Specks

In the immune system, cells are directed to the lymph node by a process of chemotaxis. In this case, dendritic cells express receptors for chemokines, which are transmitted by the lymph node. As already noted, we wish to avoid indiscriminate broadcasting of messages, thus ruling out the possibility of integration specks transmitting homing messages. However, we wish to direct the SM back to the integration specks. This is implemented by using a simple routing algorithm, based on spanning trees, which is described in detail in [5]. The algorithm utilised requires that each tissue speck stores locally the root ID, the ID of its parent in the tree and the number of hops that it is away from the root. It also requires that the routing paths are refreshed periodically, to restore any corrupted links due to potential failures of communication between specks.

4.3 Summary of the Model

In summary, integration specks send out SMs which traverse tissue specks, where they collect external and internal signals. They then return to the nearest

[1] This is an application dependent feature; candidates, currently being explored, take advantage of the processing power of individual tissue specks and include measuring variance of external signals and monitoring of internal signals such as battery power.

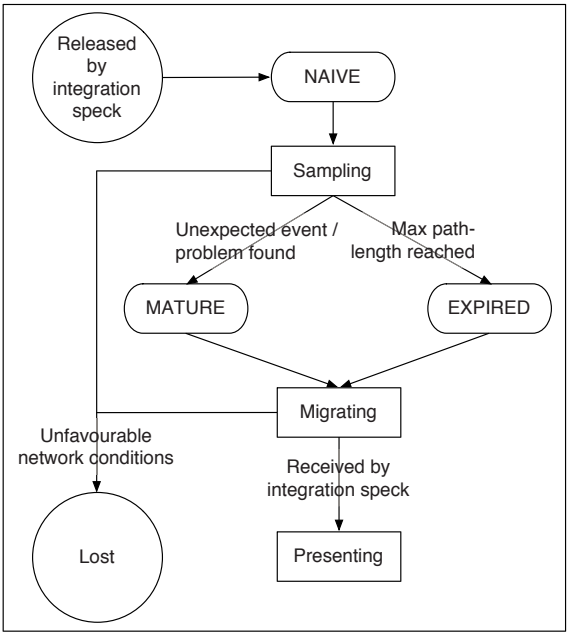


Fig. 1. The life cycle of a scouting message

integration speck, where their information is filtered and aggregated. Eventually, a decision may be made by the integration speck to act upon the collective information. This may result in one or more possible actions: effector messages may be sent out, which modify the external environment (e.g. turning a heat source on or off); alternatively, the integration speck may modify the internal variables of the system, for example alerting tissue specks to modify their internal parameters, or increasing the rate at which it sends out SMs in order to gather further information. We intend to examine these further in future work.

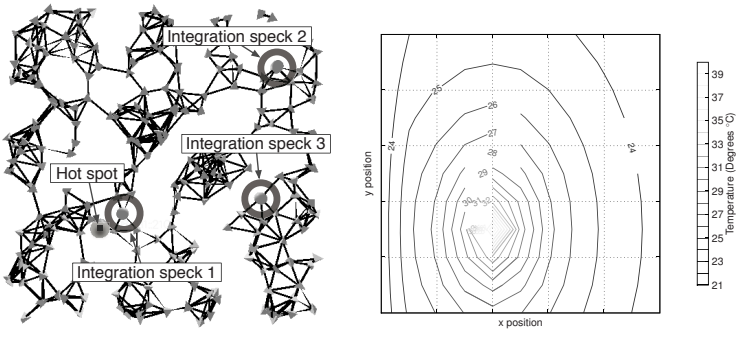
5 Verification of the Model in Realistic Topologies

5.1 SpeckSim

In the first instance, we test our model in a behavioural speck simulation environment, SpeckSim [9], provided by the Speckled Computing consortium. Details of the simulation tool can be obtained from [1]. The simulation tool has been augmented to enable the immune model to be deployed, details for which are given in [5]. In addition, a heat model was added which simulates ambient temperature in the environment. Furthermore, it allows the introduction of hot spots, which radiate heat at variable temperatures. Specks are assumed to contain temperature sensors, which incorporate both white noise and an individual tuneable random bias in the readings.

5.2 Simulation Setup

In [5] we presented preliminary results regarding the coverage of the network by SMs. These results were obtained in static Specknet deployments, in which specks were arranged in a regular grid pattern on a 2D plane. We now extend this analysis in more realistic topologies, in which specks are randomly distributed on a 2D plane. The radio range is fixed such that, on average, each speck is within communication range of four neighbours (the actual value of the range was determined empirically from extensive investigation with random topologies). Three integration specks are randomly positioned in the Specknet, as shown in Fig. 2(a).



(a) Example topology (b) Example heat map

Fig. 2. A snapshot from SpeckSim shows an example of one of the network layouts used for experiments (a). The corresponding heat map is shown in (b).

Initialisation Phase. When a simulation run is launched, devices are initialised at random times to reduce the chances of collision. After a random start-up delay, specks establish their local neighbourhoods by broadcasting their IDs. The one-hop neighbourhood list, that each speck maintains, is updated by periodic broadcasts of its own ID (further details on the formation of neighbourhoods is given in [9]). In this phase, the spanning trees are also established. Full details of this process are given in [5].

Operational Phase. Upon completion of the initialisation phase, integration specks start producing SMs, currently with fixed, pre-determined frequency. Each SM contains information regarding the number of tissue specks that the message must sample (i.e. the path-length) before it expires, the type of local samples to collect (e.g. local mean or maximum value over a specified window size), and the type of group sample to return (e.g. the mean of the local samples). This information may be altered by the integration speck as time progresses, based on the information it is currently receiving.

Using this Specknet deployment, a number of experiments are performed in which the path-length of the SMs and the topology of the network is varied, in an environment defined by the heat model. The results are presented in the next section.

6 Results

Experiments were conducted using ten, randomly generated, topologies, each containing 225 specks. Three randomly positioned specks were assigned to be integration specks, the remainder allocated as tissue specks. All specks refresh their list of neighbours every 10 time units; the lifetime of a neighbourhood record for each speck is 15.1 time units; the spanning trees are refreshed at a frequency of 5 time units — spanning tree records never expire; SMs are generated from integration specks every 2 time units. These values are currently chosen arbitrarily. In the future, extensive experimentation will be performed in order to optimise the parameter settings and determine the robustness of the system to each parameter.

6.1 Coverage of Network

To determine the relationship between the coverage of the entire Specknet and the path-length of the SMs, ten experiments were performed for each topology, in which the number of scouting messages returning to the integration specks was measured over a time period of 255 units (in which 100 SMs were sent). Path-lengths were varied systematically from 1 to 50. Figure 3 presents the results from the perspective of the number of specks sampled and the number of messages sent. The percentage of *sampled* specks refers to the percentage of all tissue specks that have been sampled by a SM at least once. On the other hand, *successfully sampled* specks is the percentage of the tissue samples taken, that are actually received by integration specks, and therefore may be lower than the former measure. Finally, we also measure the percentage of SMs that are sent but do return to an integration speck, which comprise the *lost* SMs. Loss may occur due to radio collisions or timed out broadcasts.

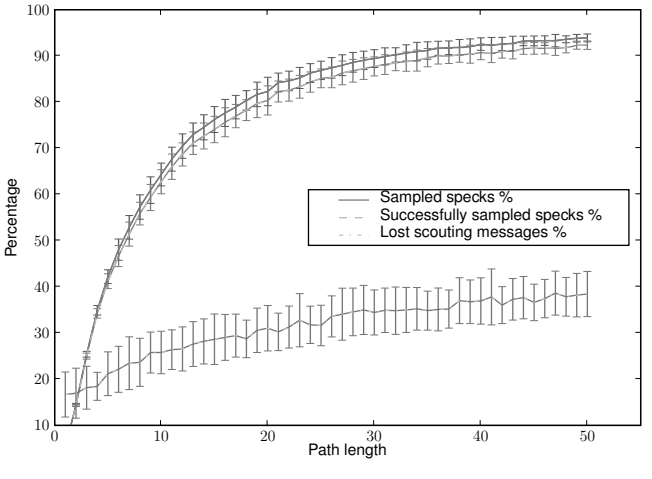


Fig. 3. Sampled specks and lost scouting messages over path-length

Figure 3 shows that at low path-lengths, transmission and reception of SMs is reliable; as the path-length increases, lost messages account for at maximum 30% of messages sent at path-length 50. On the other hand, the success-rate, as measured by the number of specks *successfully sampled*, increases with path-length; the longer path-length allows parts of the Specknet to be sampled that may not be possible to reach at short path-lengths, given the random placement of the integration specks. The figure shows that despite the loss of up to a third of messages at high path-length, almost 100% coverage of the Specknet can still be achieved. Clearly, the low success-rate at low path-length is inevitable, but could be countered by increasing the number of integration specks in the system.

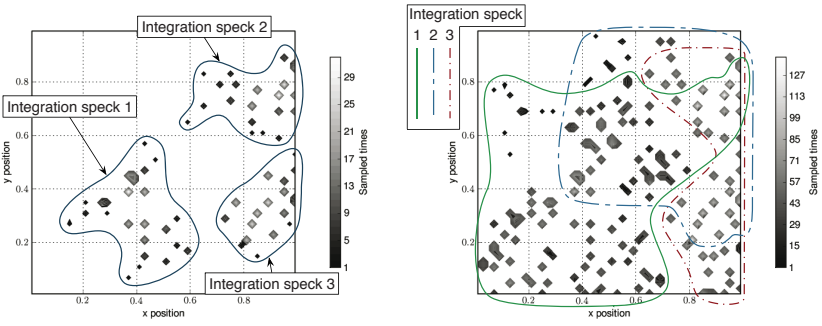
6.2 Locality of Information

The path-length of a SM plays a key role in determining how localised is the information that is returned to an integration speck. Low path-lengths result in a snapshot of the immediate environment of an integration speck being obtained. In contrast, high path-lengths result in messages scouting larger areas and a more general picture being built up. This is illustrated in Figs. 4(a) and 4(b) which depict the tissue specks sampled by each integration speck at path-lengths 5 and 35 respectively. The shading of specks reflects the number of times a speck was sampled (lighter shading indicates more samples). Blank areas indicate regions which were not sampled at all *or* do not contain any specks (the topology corresponds to that shown in Fig. 2(a)). At low path-length, each integration speck receives a distinct picture of its local environment. On the other hand, at path-length 35, the regions sampled overlap. We expect that this will have significant impact on the next stage of our research, in which the integration specks will be required to route effectors back to sites of interest, just as the lymph nodes route T cells to infected sites as reported by dendritic cells.

6.3 Monitoring Environmental Changes

In the above experiments, the heat model used maintained the ambient temperature at 21°C; tissue specks sample the temperature from the environment every time unit and maintain a window of the last 4 readings. They pass their local sample mean S_L over this time window to a SM. Each speck is assigned a random bias at the start, drawn from a Gaussian distribution with mean 0 and standard deviation 1, and a random measurement noise, again drawn from a Gaussian distribution with the same parameters.

In order to test the ability of the integration specks to monitor the local environment based on information returned by SMs, we performed a further experiment in which a hot spot was introduced into the environment at time step t=30; the temperature at this hot spot increased linearly over 40 time steps, to reach 40°C at $t = 70$. The temperature was then maintained at this value



(a) Path-length = 5 (b) Path-length = 35

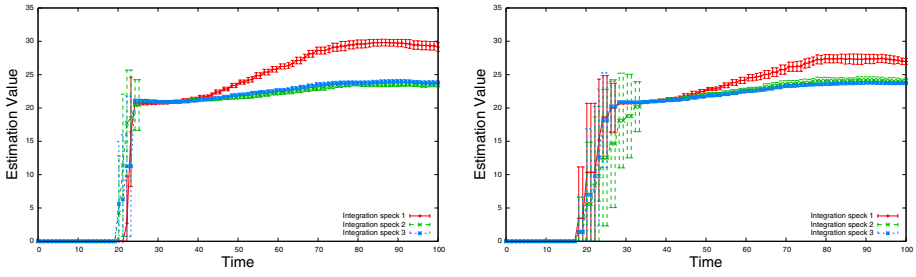
Fig. 4. Coverage of the tissue specks' network in terms of scouting messages returning to each integration speck; shading shows the number of times a speck was sampled

until $t = 80$, at which point it linearly decreased to a value of 38°C at $t = 100$. The hot spot causes a temperature gradient throughout the environment, shown in Fig. 2(b) — the hot spot itself is indicated on Fig. 2(a).

Scouting messages returning to an integration speck, notify the speck of their group sample mean S_G. The integration speck maintains a list of the last 5 temperature values delivered by SMs and calculates an estimation mean value E. In Figs. 5(a) and 5(b), we plot the estimation value E against time for each of the integration specks for path-lengths 5 and 35; the graphs shown are the result of averaging over 30 separate experiments on a single topology.

From both graphs we can see the initialisation phase of the network, which is completed in, approximately, the first 20 time units of the simulation runs. This phase is followed by a start-up transient, during which the integration specks fill their empty buffers with received SMs. At this stage, the results are inaccurate, as the integration specks have not yet collected sufficient samples for calculating their estimation values. This stage lasts for a longer period in the latter figure due to the much longer path-length that the SMs must complete, before they expire and head back to an integration speck. After this necessary time lag, the network enters the operational phase of its life.

In both cases, the temperature at each of the three integration specks stabilises at the ambient value. Furthermore, in both figures, it is clear that integration speck 1, which is closest to the hot spot, becomes aware of the temperature change in its local environment, whilst the remaining integration specks record only a slight increase in temperature. Figure 5(a) clearly shows that the short path-length results in SMs capturing a more tightly localised representation of the environment; integration speck 1 records a maximum average temperature of approximately 30°C. This is contrasted in Fig. 5(b), in which integration speck 1 registers a maximum temperature of approximately 26°C, reflecting the sampling of greater regions indicated in Fig. 4(b).



(a) Path-length = 5 (b) Path-length = 35

Fig. 5. Estimation mean temperature E obtained by integration nodes over time from returning scouting messages

7 Conclusions

The paper has presented a model inspired by the innate immune system for computing the state of an environment using a Specknet; both external environmental signals as well as internal system variables, regarding the state of the system itself, can be monitored. The experiments presented in the paper, verify the basic premises of the model when deployed in a simulated environment consisting of randomly placed specks. The results validate that SMs can be used to sense the state of the environment; experiments have also analysed the relevance of some of the parameters of the model, in particular the effect of the path-length of SMs. In future work, we plan to investigate the scaling up of network size, and analysing the effect of increasing the number of integration specks. In parallel, we are currently experimenting with refinements of the model, in which SMs mature as a result of collecting 'interesting' information, and immediately return to the integration specks. We also intend to examine how information received by SMs can be integrated and acted upon.

The Specknet environment presents an exciting, but challenging, platform for research in autonomous systems. The natural immune system formulates an immune response as a result of the cumulative experience of the immune system dealing with both the body and the world. Ee hope to achieve desired responses in a Specknet by computing the state of the Specknet and reacting accordingly.

References

1. http://www.specknet.org/dev/specksim
2. Arvind, D., Elgaid, K., Krauss, T., Paterson, A., Stewart, R., Thayne, I.: Towards an Integrated Design Approach to Specknets. In: IEEE Int. Conf. Communications 2007, ICC 2007, pp. 3319–3324 (2007)
3. Atakan, B., Akan, Ö.B.: Immune System Based Distributed Node and Rate Selection in Wireless Sensor Networks. In: BIONETICS 2006: Proc. 1st Int. Conf. Bio-Inspired Models of Network, Information and Computing Systems, p. 3. ACM Press, New York (2006)

4. Bokareva, T., Bulusu, N., Jha, S.: SASHA: Toward a Self-Healing Hybrid Sensor Network Architecture. In: Proc. 2nd IEEE Workshop Embedded Networked Sensors (EmNetS-II) (June 2005)
5. Davoudani, D., Hart, E.: Computing the State of Specknets: An Immune-Inspired Approach. In: Proc. Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2008, Edinburgh, UK, June 16–18 (2008)
6. Davoudani, D., Hart, E., Paechter, B.: An Immune-Inspired Approach to Speckled Computing. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 288–299. Springer, Heidelberg (2007)
7. Dressler, F.: Benefits of Bio-inspired Technologies for Networked Embedded Systems: An Overview. In: Dagstuhl Seminar 06031 on Organic Computing - Controlled Emergence, Schloss Dagstuhl, Wadern, Germany (January 2006)
8. Greensmith, J., Aickelin, U., Twycross, J.: Articulation and Clarification of the Dendritic Cell Algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 404–417. Springer, Heidelberg (2006)
9. McNally, R., Arvind, D.K.: A Distributed Leaderless Algorithm for Location Discovery in Specknets. In: Kermarrec, A.-M., Bougé, L., Priol, T. (eds.) Euro-Par 2007. LNCS, vol. 4641. Springer, Heidelberg (2007)
10. Sompayrac, L.: How the Immune System Works, 3rd edn. Blackwell Publishing, Malden (2008)

A Hybrid Model for Immune Inspired Network Intrusion Detection

Robert L. Fanelli

Department of Information and Computer Science
University of Hawaii at Manoa
Honolulu, HI 96822 USA
fanelli@hawaii.edu

Abstract. This paper introduces a hybrid model for network intrusion detection that combines artificial immune system methods with conventional information security methods. The Network Threat Recognition with Immune Inspired Anomaly Detection, or NetTRIIAD, model incorporates misuse-based intrusion detection and network monitoring applications into an innate immune capability inspired by the immunological Danger Model. Experimentation on a prototype NetTRIIAD implementation demonstrates improved detection accuracy in comparison with misuse-based intrusion detection. Areas for future investigation and improvement to the model are also discussed.

1 Introduction

Preserving the confidentiality, integrity and availability of networked systems is an increasingly important and difficult task. Misuse-based network intrusion detection systems have been an effective safeguard against known threats to networked assets. Improving these systems with adaptive capabilities to detect novel threats and with improved accuracy to reduce false alarms will maintain their usefulness.

A long-time goal of the security community has been to create an 'immune system' for information systems with the flexibility, effectiveness and robustness of the immune systems that protect organisms [7]. A system that responds effectively to new threats without human intervention would significantly improve security.

Artificial immune systems (AIS) offer a means to solve complex, dynamic problems like many of those found in the domain of information system security [5]. However, problems of scalability and detection of a broad range of potential threats have so far limited the success of intrusion detection systems based solely on artificial immune systems.

A hybrid threat detection model that combines artificial immune system methods with conventional intrusion detection techniques has the potential to provide results superior to that offered by either of these approaches separately. Such an approach could be a step toward more secure, self-protecting information systems.

This paper presents a model for network threat recognition with immune-inspired anomaly detection. This model combines immune-inspired mechanisms with proven, conventional network intrusion detection and monitoring methods. The model builds

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 107–118, 2008.

upon these conventional methods, employing them as the foundation for an innate immune capability rather than replacing them. This approach preserves the functionality of an existing set of information security applications, providing enhanced capabilities via the artificial immune system overlay.

The remainder of this paper is organized as follows. Section 2 provides background information on the biological inspiration for the model and a brief overview of related work. Section 3 presents the model, covering the components, representation schemes and methods of operation. Section 4 relates experimentation on a prototype implementation of the model. Section 5 discusses conclusions and areas for future work.

2 Background

2.1 Immunological Inspiration

The Danger Model is an alternative to self - nonself discrimination as an explanation for behavior of the immune system [19]. This theory states that the primary trigger of an immune response is not the fact that a pathogen is foreign so much as that it does harm and is therefore dangerous. The Danger Model holds that the body's tissues, not the immune cells, are the primary controllers of the immune response [21]. Distressed tissues emit chemical danger signals to stimulate immune reactions while healthy tissues emit 'calming' or safe signals to induce tolerance by the immune system [20].

Antigen presenting cells such as dendritic cells exist in tissues throughout the body as part of the innate immune system. These cells spend a time in an immature state during which the sample their surroundings to collect antigens. Eventually a dendritic cell matures, leaving the tissue and migrating to the lymphatic system. Once there, dendritic cells present their antigens to the adaptive immune system's T cells.

The Danger Model theorizes that dendritic cells mature due to stimulation from danger and safe signals. Further, maturation in a context of danger will cause the dendritic cell to signal that the antigens presented require an immune reaction while maturation in a context of safety signals that the antigens should be tolerated.

2.2 Related Work

Several authors apply AIS methods to problems in network intrusion detection. Kim et al. provide a detailed review of this work [16]. Hofmeyr and Forrest present an implementation of a network intrusion detection system called the Lightweight Intrusion Detection System, or LISYS [13]. LISYS uses distributed populations of negative detectors to identify anomalous TCP SYN connections on a single network broadcast domain. Kim and Bentley propose a model for an immune-inspired network intrusion detection system [14]. They subsequently present a dynamic clonal selection algorithm, DynamiCS, that attempts to overcome scaling issues in the LISYS approach [15].

Aickelin and Cayzer discuss applying Danger Model concepts to AIS [2]. Aickelin et al. subsequently argue that the Danger Model provides a good source of inspiration for AIS that address intrusion detection problems [1]. They describe a framework for intrusion detection based on the correlation of danger signals derived from the state of systems and observed events, with the type and strength of the signals determining the occurrence of alerts or other reactions. Kim et al. present a Danger Model inspired

approach for host-based intrusion detection [17] while Tedesco, Twycross and Aickelin address network intrusion detection [25]. These approaches emulate the interactions of dendritic cells and T cells to identify and respond to pathogens. Dendritic cells stimulate or suppress the reaction of T cells to a given antigen based on the presence or absence of danger signals with the antigen. Twycross and Aickelin propose a larger role for innate immune system concepts in AIS and provide the *libtissue* software framework to facilitate implementation of these concepts [26, 27].

Greensmith, Aickelin and Cayzer discuss fully incorporating Danger Model concepts into an actual AIS anomaly detector by emulating the functioning of dendritic cells [10]. This proposal has been developed into the Dendritic Cell Algorithm (DCA) that demonstrates promising results on a number of problems, including network port scan detection [11, 12].

3 A Model for Immune Inspired Intrusion Detection

The Network Threat Recognition with Immune-Inspired Anomaly Detection (NetTRIIAD) model draws inspiration from both the innate and adaptive portions of the natural immune system. The model can be logically divided into an Innate Layer and an Adaptive Layer. Figure 1 depicts an overview of the NetTRIIAD model.

The Innate Layer conducts the majority of NetTRIIAD's external data collection. This layer synthesizes antigens from packets observed on the network. It also synthesizes danger model signals from observed events and the state of the network and its hosts. The Innate Layer classifies antigens as dangerous or safe and provides this information the Adaptive Layer for further processing.

The Adaptive Layer emulates the interactions that occur between the adaptive immune system's T cells and dendritic cells in locations such as the paracortex of a lymph node. This layer processes the antigens presented by dendritic cells migrating from the Innate Layer. The Adaptive Layer recognizes threats visible on the network, using a combination of self - nonself discrimination on the presented antigens and the Innate Layer's classification of the antigens as dangerous or safe.

3.1 Representation Schemes

The NetTRIIAD model uses two primary structures to represent information: antigens and danger model signals. The antigens represent network traffic, with each observed packet resulting in the synthesis of a corresponding antigen. The NetTRIIAD antigen contains two types of features: address features and protocol features. *Address features* are 32-bit, unsigned integer values corresponding to the network-ordered representation of an Internet Protocol, version 4 (IPv4) address. *Protocol features* are 32-bit, unsigned integer values derived from the protocol value found in the IPv4 packet header and, for Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) packets, a port value. The value of a protocol feature is: (IP protocol value * 65536) + port value. A NetTRIIAD *antigen* is then a vector of four real-valued features derived from an Internet Protocol Version 4 (IPv4) packet: *destination_address*, *source_address*, *destination_protocol*, and *source_protocol*.

NetTRIIAD *danger model signals* emulate the various chemical signals that promote and suppress reactions in the immune system. A danger model signal includes two functional elements. First is a single *feature value* that affects the signal's binding potential. This is an address or protocol feature value in the same format as the features that comprise antigens, described above. The other functional element is a *signal level* value. This is an integer value that determines the degree of danger or safety the signal represents. A danger model signal with a signal level that indicates danger is a *danger signal*. Similarly, a danger model signal with a signal level that indicates safety is a *safe signal*.

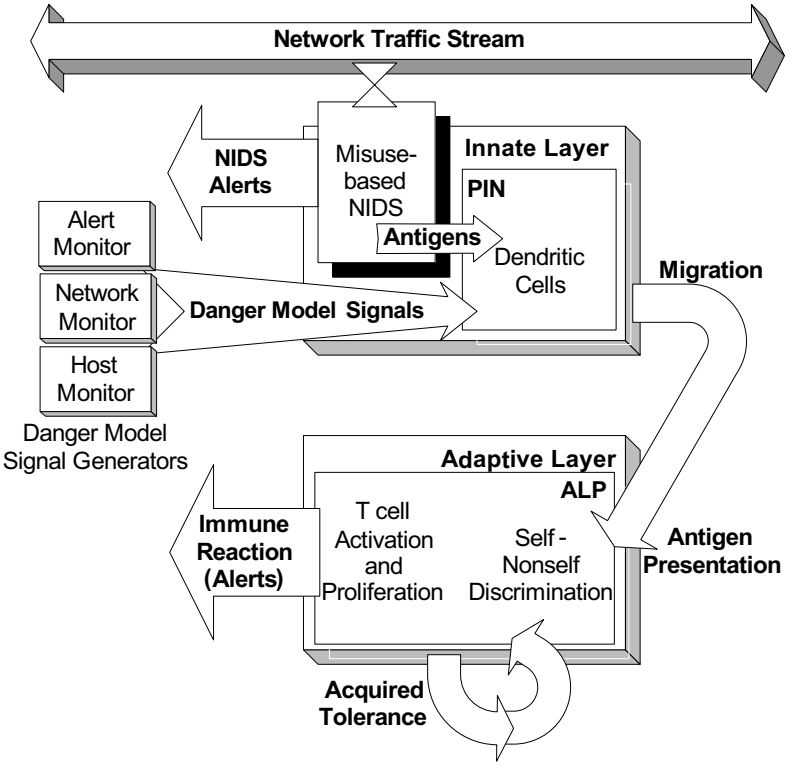


Fig. 1. NetTRIIAD Model Overview

3.2 The Innate Layer

The NetTRIIAD Innate Layer emulates the functions of the dendritic cells in the tissues of an organism. The Innate Layer consists of a misuse-based network intrusion detection system (NIDS), a set of danger model signal generators, and a Peripheral Immune Node in which the artificial dendritic cells, antigens and danger model signals interact.

The misuse-based NIDS component gives NetTRIIAD an innate ability to recognize known network threats. A NIDS *alert event* occurs when network traffic matches an element of the rule set. An alert event includes, as a minimum, both destination

and source IP addresses and a labeling of the type of alert. The alert event may also include protocol and other amplifying information.

The *danger model signal generators* process external data pertaining to the state of the network and its attached systems, emitting corresponding danger model signals. Each generator monitors a specific set of external data, synthesizing danger model signals and forwarding them to the Peripheral Immune Node. NetTRIIAD includes three danger model signal generators: the Alert Monitor, Host Monitor and Network Monitor.

The *Alert Monitor* observes the alert events emitted by the NIDS and emits a corresponding sequence of danger model signals. Evidence of danger seen in the network traffic can thus affect the immune response. This is inspired by the ability of the innate immune system to detect the presence of pathogens through its sensitivity to various pathogen associated molecular patterns (PAMP).

The Alert Monitor synthesizes two danger model signals for each alert event observed. One signal has its feature value set to that of the alert event source address while the other has its feature value set to the destination address. The Alert Monitor sets the signal level of the danger model signals based upon the type of the alert event. The Alert Monitor contains a mapping of each of the possible alert types to a specific danger model signal level. For example, an alert event indicating a more serious threat, such as attempted root access, maps to a correspondingly higher danger value.

The *Host Monitor* emits danger model signals corresponding to the perceived 'health' of a population of hosts. This allows the state of the hosts to affect NetTRIIAD in a manner analogous to the effect of tissue states on the immune response. Indications of damage promote immune reactions while indications of normal operation, or 'health', suppress immune reactions.

The Host Monitor periodically retrieves status information about each host in the monitored population. The sampling interval is measured by the network traffic flow, with a status sample being retrieved each time ℓ_h packets are observed. Each Host is classified into one of four states based upon the status information received and assigned a corresponding danger or safe signal level. The Host Monitor emits a danger model signal for each host with a determined state during each sampling interval. The feature value of each signal is determined by the IP address of the corresponding host and the signal level follows from the host's state. Table 1 describes the host state criteria and resulting signal levels.

Table 1. Host State Classification

Host State	***Host Status***	***Resulting Signal***
Necrotic	The host is non-operational	Full strength danger signal ($s = b_n$)
Stressed	The host is operating but is impaired	Reduced strength danger signal ($s = b_n / 2$)
Healthy	The host is operating normally	Safe signal ($s = c_n$)
Undetermined	Host is in scheduled down time or status can not be determined	No signal

The *Network Monitor* observes the overall state of the network traffic flow, emitting corresponding danger model signals. This emulates the effect of general stress signals on the immune response. Tissues under stress emit chemical signals that promote immune reactions while unstressed tissues suppress immune reactions [22]. The Network Monitor considers four network traffic parameters to determine the network stress level: the overall traffic rate (X_r), the rate of ICMP unreachable packets (X_u), the rate of TCP SYN packets (X_s) and the percentage of packets dropped by the misuse-based NIDS (X_d).

The first three parameters are commonly employed to detect undesirable network activity [6]. The fourth parameter, X_d, is a useful indicator of stress in network intrusion detection scenarios [3]. The use of network statistics, such as packet rates, for the synthesis of danger signals has also been explored in related work [13].

The Network Monitor makes an observation of the network parameters after each interval of ℓ_n observed packets, and then compares the observation with a *traffic parameter profile*. This profile consists of a separate mean and standard deviation value for each of the four parameters, computed from previous observations. The Network Monitor incorporates each observation into the traffic parameter profile, allowing the stress computations to adapt to routine parameter changes over time.

The Network Monitor computes an individual *stress value* for each of the four parameters, using the observed value and the mean and standard deviation.

$$stress\_value(i) = \begin{Bmatrix} 0 & if & X_i - \mu_i \le \delta_i \\ \dfrac{X_i - \mu_i - \delta_i}{2\delta_i} & if & \delta_i < X_i - \mu_i < 3\delta_i \\ 1 & if & X_i - \mu_i \ge 3\delta_i \end{Bmatrix} \quad (1)$$

Where X_i is the observed value, μ_i is the mean and δ_i is the standard deviation for parameter i. The overall stress level is the arithmetic mean of the four stress values:

$$stress = \frac{\left(stress\_value(r) + stress\_value(u) + stress\_value(s) + stress\_value(d)\right)}{4} \quad (2)$$

The Network Monitor then emits a single, general danger model signal with a 'wildcard' feature value that allows it to bind with every artificial dendritic cell. If the stress value is less than the *safe stress level threshold*, t_n, the signal is safe signal with strength determined by:

$$s = c_n \times \left(1 - \frac{stress}{t_n}\right) \quad (3)$$

Otherwise the signal is a danger signal with strength determined by:

$$s = b_n \times \left(\frac{stress}{(1 - t_n)}\right) \quad (4)$$

Where b_n is the danger signal coefficient, c_n is the safe signal coefficient and s is the resulting signal strength.

The *Peripheral Immune Node* (PIN) provides the location in which NetTRIIAD antigens and danger model signals interact to classify antigens, and thus the corresponding packets, as 'dangerous' or 'safe'. The PIN contains a population of artificial dendritic cells to carry out the classification process. The PIN takes as its input the antigens corresponding to observed packets and the signals emitted by the danger model signal generators. Its output is a stream of artificial dendritic cells that present antigens and the corresponding contexts of danger or safety in which they were collected.

An *artificial dendritic cell* (DC) acts as a container for an antigen and corresponding danger model signals. Each DC presents exactly one antigen while no antigen is presented by more than one DC at a time. The DC contains a *danger level* value that is the sum of the signal strengths of all the danger signals that have bound with the DC. Similarly, the DC also contains a *safe level* value for the sum of binding safe signals. Finally, the DC maintains an *antigen count* that records the number of antigens that have arrived since the DC was instantiated and match the DC's antigen.

When a danger model signal arrives, the PIN compares the signal with each DC to determine if binding occurs. A danger model signal binds with a DC if the danger model signal's feature value is equal to any of the feature values in the antigen presented by the DC. A binding danger model signal adds its signal level to the DC's danger level or safe level, as appropriate. Binding does not exhaust or otherwise alter a danger model signal. A single danger model signal arriving at the PIN may thus bind with and stimulate multiple DCs.

DCs persist in the PIN until they mature. A DC matures in a *dangerous context* when its danger level reaches the *danger maturation threshold.* Similarly, the DC matures in a *safe context* if its safe level reaches the *safe maturation threshold*. The DC leaves the PIN and migrates to the Adaptive Layer to present its antigen and the context in which it matured.

3.3 The Adaptive Layer

The Adaptive Layer emulates the interactions that occur between dendritic cells andthe adaptive immune system's T cells in locations such as the paracortex of a lymph node. The Adaptive Layer identifies threats through immune reactions triggered by the activation and proliferation of artificial T cells. This emulates the action of helper T cells stimulating immune reactions in the natural immune system to specific antigens identified as threats.

The main component of the Adaptive Layer is the *Artificial Lymphatic Paracortex* (ALP). The ALP contains populations of *artificial T cells* (TC), each of which represents a population of identical, activated T cells. The TC contains a *T cell receptor* (TCR), a vector of values defined identically to the NetTRIIAD antigen, specifying which antigen will bind to the TC. The TC maintains a *population* value for the quantity of T cells it represents. It also records the sum of the antigen count values of the DCs that have presented antigens to the TC. TCs exist in two classes: *effectors* that promote immune reactions to antigens matching their TCR and *regulators* that suppress immune reactions.

NetTRIIAD does not attempt to explicitly represent the huge population of naïve, nonself-reactive T cells to implement self – nonself discrimination (SNSD). NetTRIIAD uses a positive characterization approach that explicitly defines a set of self antigens. The self set contains antigens representing the network's normal, threat-free traffic. An antigen is classified as self if and only if it is found to exist in this self set, otherwise it is nonself. The presence of a matching naïve T cell is implied by the classification of an antigen as nonself. This avoids the potentially high overhead of generating negative detectors or other negative characterizations of self [24].

Since client / server communications frequently use dynamically assigned, or ephemeral, ports for the client endpoint, a series of essentially identical communications can result in large number of self antigens differing only in one protocol feature value. NetTRIIAD defines *protocol feature similarity classes* that specify ranges of port values that will all be considered equal if used for the client endpoint of a known client / server session. This allows a single self antigen to provide a summarized representation of multiple antigens pertaining to equivalent client / server communications, significantly reducing the size of the self set.

The ALP manages DCs and TCs using a *population update cycle* consisting of 12 *processing intervals* of eight minutes each. Thus the total population update cycle time is equal to 96 minutes, giving 15 cycles in a 24 hour period. This duration provides an adequate time window to process the antigens and danger model signals that may result from multi-staged or stealthy threats. It is also short enough to ensure TCs do not persist so long that they combine evidence from unrelated events to promote unwarranted immune reactions and cause false positive alerts [4, 29].

The ALP processes DCs as they arrive throughout the processing interval. The ALP first uses the SNSD mechanism to classify a DC's antigen as self or nonself. A self antigen is tolerated and thus receives no further processing. A nonself antigen results in activation and proliferation of a TC with matching TCR. A DC presenting in the dangerous context results in a TC of the effector class with a population equal to the DC's danger level. Conversely, presentation in the safe context results in a TC of the regulator class with a population equal to the DC's safe level. The new TC's antigen count is equal to the antigen count of the presenting DC. If a TC of the same class and TCR value already exists in the ALP, the new TC merges with it, increasing the total population and antigen count of the existing TC. Otherwise the new TC joins the ALP population directly.

At the conclusion of each processing interval, the ALP checks for immune reactions and updates the TC population. The ALP determines the effect of immune suppression by calculating a net population for each effector TC. The *net population* is the effector TC population minus the population of the regulator TC with a matching TCR, if such exists.

The ALP carries out a *clonal selection* process on the effector TCs to ensure only those most representative of a given threat will promote an immune reaction. The clonal selection has the effect of clustering the TCs corresponding to a given group of traffic sources and destinations and eliminating as potential false positives those stemming from weaker danger signals. The ALP partitions the TC population into disjoint sets using the address features in their TCRs [8]. Two TCs fall in the same disjoint set if the same address feature value appears in both their TCRs. The ALP determines the maximum net population value among the TCs in each disjoint set. Any

TCs in the disjoint set having a net population value less than one half this maximum value receives no further consideration for immune reaction computations in the current interval.

The ALP evaluates the remaining TCs to determine if any triggers an immune reaction. A TC's *danger concentration* is the net population divided by the TC's antigen count value. This gives greater weight to the antigens, and thus packets, that appear less frequently in a given dangerous context. If the danger concentration exceeds the *immune reaction threshold*, the TC is considered to have caused an immune reaction to the antigens matching its TCR. The ALP emits a threat alert, using the IP address, protocol and port information contained in the TCR.

TCs do not persist in the ALP indefinitely. At the conclusion of each processing interval, the ALP carries out a *population decay* operation that reduces the population level of each TC by one half. A TC with a population value decaying to zero is eliminated. This gives each T cell a finite lifespan and causes the effect of each antigen presentation by a DC to fade away over time.

The ALP also includes an *acquired tolerance* mechanism that allows it to adapt to changes in the normal traffic on the network, adding self antigens for traffic found to be safe. If a regulator class TC persists in the ALP for a complete population update cycle during which no immune reaction to the antigen matching its TCR occurs, the TC causes a *tolerization reaction*. The ALP acquires a new self antigen corresponding to the regulator TC's TCR and will henceforth classify this antigen as self and thus tolerate it. Since population decay constantly reduces the TC population, strong evidence in the form of multiple safe presentations of the same antigen are needed for a regulator TC to persist long enough to cause a tolerization reaction.

4 Experimentation

Experimentation on a prototype implementation of the NetTRIIAD model provides a comparison of its performance versus that of a conventional NIDS.

The NetTRIIAD components are implemented as separate processes, with socket-based inter-process communication enabling flexible deployment in support of typical NIDS architectures. The prototype builds upon proven, conventional information security tools, augmenting their capabilities. The prototype uses the Snort NIDS, version 2.6.1.3 as the misuse-based NIDS component [23]. A custom Snort plug-in synthesizes NetTRIIAD antigens from captured packets and retrieves network statistics for the Network Monitor. The prototype also uses the Nagios network monitoring application to gather information for the Host Monitor [9].

The experimentation compares the detection results of NetTRIIAD with those of a baseline Snort installation on the DARPA / MIT 1999 Intrusion Detection Evaluation (IDEVAL99) inside data sets [18]. Host status changes required to drive the Host Monitor process were reconstructed from the IDEVAL99 documentation and played back through Nagios in synchronization with the recorded network traffic. The NetTRIIAD self set was captured from two weeks of threat-free IDEVAL99 training data. The 16 million packets in this training data yielded 31, 215 distinct self antigens.

Both installations ran against the two weeks of inside evaluation data with true and false positive detections determined by comparison of the alert output with the

IDEVAL99 master identifications list. Figure 2 depicts the true positive rates and positive predictive value figures from the experimental runs. There were no significant differences in the true positive rates ($t = 1.01766$, $df = 16$, $p > .25$). However, the NetTRIIAD implementation had a significantly lower number of false positive detections. This led to a positive predictive value of 0.65, significantly better than the 0.38 returned by Snort alone ($t = 4.85328$, $df=16$, $p < .001$).

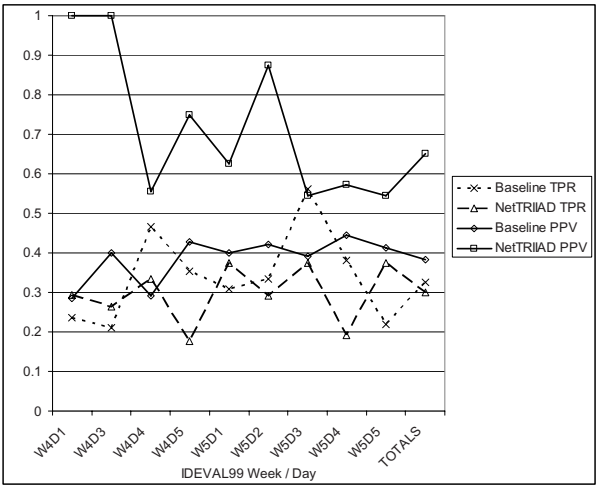


Fig. 2. Performance of NetTRIIAD and Snort Baseline

5 Conclusions and Future Work

NetTRIIAD demonstrates performance improvements over a conventional, misuse-based network intrusion detection system. NetTRIIAD shows a significantly better positive predictive value for threat detection than is achieved by a conventional misuse-based NIDS on the same data. A high positive predictive value has two benefits for protecting networks. One is as a 'priority of work' mechanism for security analysts, allowing limited resources to be focused on actual threats. The other benefit is in facilitating intrusion prevention and other automated security responses. Any system empowered to automatically respond to threats runs the risk of doing more harm than good through disruption of legitimate network traffic unless it can accurately differentiate true threats from false alarms.

The NetTRIIAD model builds upon trusted information security tools, preserving their effectiveness while providing improved performance with the addition of immune inspired components.

Several opportunities for future work exist. The initial NetTRIIAD experimentation occurred in a controlled laboratory environment as an essential first step in developing a usable security tool from the model. However, further experimentation on live networks would help to better understand the true utility of NetTRIIAD.

Extending NetTRIIAD to move beyond threat recognition and include automated threat response would be a step towards a computer immune system and would benefit from the improved positive predictive value. The features in a NetTRIIAD antigen contain sufficient information to create firewall rules to block or shape the associated traffic. Such work could extend the adaptive immune metaphor beyond T cell activation, adding elements inspired by B cells and antibody production.

Improvements to the danger model signal generators, possibly to examine additional external data sources, could gather better evidence of threats and improve detection. Similarly, a mechanism for accurate reactions to 'dangerous self', suggested by the Danger Model, would permit NetTRIIAD to recognize threats hidden in 'normal' network traffic.

References

1. Aickelin, U., Bentley, P., Kim, J., McLeod, J., Cayzer, S.: Danger Theory: The Link between AIS and IDS? In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
2. Aickelin, U., Cayzer, S.: The Danger Theory and Its Application to Artificial Immune Systems. In: 1st International Conference on Artificial Immune Systems (ICARIS 2002), University of Kent, Canterbury, UK, pp. 141–148 (2002)
3. Beale, J., Caswell, B., Kohlenberg, T., Poor, M.: Snort 2.1 Intrusion Detection. Syngress Publishing, Rockland (2004)
4. Chen, B.C., Yegneswaran, V., Barford, P., Ramakrishnan, R.: Toward a Query Language for Network Attack Data. In: 22nd International Conference on Data Engineering Workshops (ICDEW 2006), pp. 28–36. IEEE Press, New York (2006)
5. DeCastro, L., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
6. Deri, L., Suin, S., Maselli, G.: Design and Implementation of an Anomaly Detection System: an Empirical Approach. In: TERENA Network Conference, Zagreb, Croatia (2003)
7. Forrest, S., Hofmeyr, S.A., Somayaji, A.: Computer immunology. Communications of the ACM 40(10), 88–96 (1997)
8. Galil, Z., Italiano, G.F.: Data structures and algorithms for disjoint set union problems. ACM Computing Surveys 23(3), 319–344 (1991)
9. Galstad, E.: Nagios Home Page (2007), http://www.nagios.org
10. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
11. Greensmith, J., Aickelin, U., Twycross, J.: Articulation and Clarification of the Dendritic Cell Algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 404–417. Springer, Heidelberg (2006)
12. Greensmith, J., Twycross, J., Aickelin, U.: Dendritic Cells for Anomaly Detection. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006). IEEE Press, Vancouver (2006)
13. Hofmeyr, S.A., Forrest, S.: Architecture for an Artificial Immune System. IEEE Transactions on Evolutionary Computation 8(4), 443–473 (2000)
14. Kim, J., Bentley, P.: An Artificial Immune Model for Network Intrusion Detection. In: 7th European Congress on Intelligent Techniques and Soft Computing. Aachen (1999)

15. Kim, J., Bentley, P.J.: Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection. In: IEEE Congress on Evolutionary Computation (CEC 2001), pp. 1244–1252. IEEE Press, New York (2002)
16. Kim, J., Bentley, P.J., Aickelin, U., Greensmith, J., Tedesco, G., Twycross, J.: Immune system approaches to intrusion detection – a review. Natural Computing 6(4), 413–466 (2007)
17. Kim, J., Greensmith, J., Twycross, J., Aickelin, U.: Malicious Code Execution Detection and Response Immune System Inspired by the Danger Theory. In: Adaptive and Resilient Computing Security Workshop, Santa Fe, NM (2005)
18. Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K.: The 1999 DARPA off-line intrusion detection evaluation. Computer Networks 34, 579–595 (2000)
19. Matzinger, P.: Tolerance, Danger, and the Extended Family. Annual Review of Immunology 12, 991–1045 (1994)
20. Matzinger, P.: The Danger Model in Its Historical Context. Scandanavian Journal of Immunology 54, 4–9 (2001)
21. Matzinger, P.: The Danger Model: A Renewed Sense of Self. Science 296, 301–305 (2002)
22. Matzinger, P.: Friendly and dangerous signals: is the tissue in control? Nature Immunology 8(1), 11–13 (2007)
23. Snort: Snort - The Open Source Network Intrusion Detection System (2007),
24. http://www.snort.org
25. Stibor, T., Timmis, J., Eckert, C.: On the Appropriateness of Negative Selection Defined Over Hamming Shape-Space as a Network Intrusion Detection System. In: IEEE Congress on Evolutionary Computation (CEC 2005), pp. 995–1002. IEEE Press, New York (2005)
26. Tedesco, G., Twycross, J., Aickelin, U.: Integrating Innate and Adaptive Immunity for Intrusion Detection. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 193–202. Springer, Heidelberg (2006)
27. Twycross, J., Aickelin, U.: Towards a Conceptual Framework for Innate Immunity. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 112–125. Springer, Heidelberg (2005)
28. Twycross, J., Aickelin, U.: Libtissue - Implementing Innate Immunity. In: IEEE Congress on Evolutionary Computation (CEC 2006), pp. 499–506. IEEE Press, New York (2006)
29. Yegneswaran, V., Barford, P., Ullrich, J.: Internet Intrusions: Global Characteristics and Prevalence. In: ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 138–147. ACM Press, New York (2003)

Credit Card Fraud Detection with Artificial Immune System

Manoel Fernando Alonso Gadi[1,2], Xidi Wang[3], and Alair Pereira do Lago[1]

[1] Departamento de Ciência de Computação
Instituto de Matemática e Estatística
Universidade de São Paulo
05508-090, São Paulo, SP, Brazil
+55 11 3091-6135
[2] Grupo Santander, Abbey National plc, Milton Keynes, United Kingdom
[3] Citibank, São Paulo, Brazil
manoel.gadi@abbey.com, xidi.wang@citi.com, alair@ime.usp.br

Abstract. We apply Artificial Immune Systems(AIS) [4] for credit card fraud detection and we compare it to other methods such as Neural Nets(NN) [8] and Bayesian Nets(BN) [2], Naive Bayes(NB) and Decision Trees(DT) [13]. Exhaustive search and Genetic Algorithm(GA) [7] are used to select optimized parameters sets, which minimizes the fraud cost for a credit card database provided by a Brazilian card issuer. The specifics of the fraud database are taken into account, such as skewness of data and different costs associated with false positives and negatives. Tests are done with holdout sample sets, and all executions are run using Weka [18], a publicly available software. Our results are consistent with the early result of Maes in [12] which concludes that BN is better than NN, and this occurred in all our evaluated tests. Although NN is widely used in the market today, the evaluated implementation of NN is among the worse methods for our database. In spite of a poor behavior if used with the default parameters set, AIS has the best performance when parameters optimized by GA are used.

1 Introduction

In recent years many *bio-inspired* algorithms are sprouting for solving the classification problems as one can see for instance in [3]. In 1998, Neal et al. [9] developed an artificial immune system (AIS), JISYS, applied it for mortgage fraud detection, and reported some first results, still based on simulated data. In 2002, the journal Nature [10] published an article on AIS where it indicated that AIS had many kinds of applications, including the detection of fraudulent financial transactions. Even though this article previewed a possible commercial application for 2003 by a British company, we are not aware of any subsequent publication on AIS in financial fraud detection which reported good experimental results. The current paper reports our studies and application of AIS on credit card fraud detection. Moreover, in contrast to the poor performance of AIS with the default parameters, we report here an optimized and robust set of

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 119–131, 2008.

parameters under which AIS led to the best results, even when compared to the best results from all other analyzed methods.

The lack of publicly available database has been a limiting factor for the publications on financial fraud detection [14], particularly credit card transactions. In fact, only few publications on this field bring a real contribution based on experiments. For instance, the method AdaCost [16,6] was developed from Adaboost [15] for credit card fraud detection, and resulted in the metaheurists Cost Sensitive [5], which can be applied for many applications where there are different costs for false positive and false negative. Comparative studies between Neural Networks (NN) and Bayesian Networks (BN) in credit card fraud detection were reported [12], which favored the result of BN.

In this paper, we present our studies of AIS compared to other techniques such as BN and NN as well. In addition, we have also included comparative studies with two other methods: Decision Trees (DT) and Naive Bayes (NB). Moreover, we take into account the skewed nature of the dataset, the different costs for false positive and false negative in order to evaluate a classifier performance, as well as the need of a parametric adjustment in order to obtain the best results for every compared method.

Background: Fraud prevention is interesting for financial institutions. The advent of new technologies as telephone, automated teller machines (ATMs) and credit card systems have amplified the amount of fraud loss for many banks. Analyzing whether each transaction is legitimate or not is very expensive. Confirming whether a transaction was done by a client or a fraudster by phoning all card holders is cost prohibitive if we check them in all transactions. Fraud prevention by automatic fraud detections is where the well-known classification methods can be applied, where pattern recognition systems play a very important role. One can learn from past (fraud happened in the past) and classify new instances (transactions). In credit card business today, perhaps the most commonly used technique is Neural Networks, for example in Fair Isaac's Falcon software as claimed in its website (http://www.fairisaac.com/fic/en/product-service/product-index/falcon-fraud-manager/). In general, the NN implementation is inside a complex work-flow system which is integrated with the bank database. When a new transaction comes in, the work-flow calculates all the input variables and outputs a fraud score. Then this score is used to decide which transaction is going to be checked manually and to order its priority.

Skewed data and other discussions: Fraud detection model is among the most complicated models used for the credit card industry. Skewness of the data, search space dimensionality, different cost of false positive and false negative, durability of the model and short time-to-answer are among the problems one has to face in developing a fraud detection model. In this article we focus our attention on skewness of the data by comparing five methods[1] .

[1] The problem of taking into account the different cost between false positive and false negative during the training phase needs a special investigation which is what we intend to conclude before December this year. The durability and short time-to-answer problem we intend to start to analyze next year.

Fraud Tagging: We have obtained our database from a large Brazilian bank, with registers within time window between Jul/14/2004 through Sep/12/2004. Each register represents a credit card authorization, with only approved transactions excluding the denied transactions. One applies the following rule for classifying an authorization: a transaction is considered *fraudulent* if, in the next 2 months after the date of the transaction, which is called performance period, either the client queried the transaction, or the bank distrusts it as a legitimate transaction and confirms it does not belong to the client; otherwise the transaction is tagged as *legitimate*. When an authorization is tagged as fraudulent[2], the Bank has almost 100% of certainty about this claim, but when the transaction is tagged legitimate, it cannot be affirmed this is in fact legitimate, but it can only be sure that the transaction was still not identified as fraudulent in the performance window. However, according to the Bank, at least 80% of the occurred frauds are identified as fraudulent in 2-month period.

Sampling: The sampling of transactions is done in two steps: first, one randomly samples card numbers to be analyzed in this period, irrespective to whether the card had or not a fraud transaction in the historical period; second, there is a weighted sampling of the class where 10% of legitimate transactions are selected and 100% fraudulent transactions are selected.

In the end, the database that we have received from the bank contains 41647 registers, from which 3.74% are fraudlent.

Categorization: We preprocess the database in three steps:

1. We apply statistical analysis in order to remove variables that are considered unimportant for the modeling (ex: card number). From 33 variables in the beginning we had 17 independent variables and 1 dependent variable (flag_fraud) after this phase;
2. We bind the variables. All variables but Merchant Category Code (MCC)[3] are categorized in at most 10 groups, one digit only. See Table 1.
3. We generate 9 splits (also known as samples) from the databases. Each split contains a pair of databases: 70% of transactions for development (training set), and 30% of transaction for validation (testing set, holdout sample). Table 2 shows that these splits have about the same number of frauds and legitimates transactions.

All 9 splits are subsequently converted to Weka [18] format (.arff), on which our studies are executed. The software Weka-3-4-11 is used for all of our studies and the implementations used for DT, BN, NB and NN are built in Weka. The only plugged in implementation was the AIS, the AIRS2 version 1.6 (March 2006) implemented by Jason Brownlee [1], originally designed by Watkins et al. [17].

[2] According to the scope of the annotated dataset provided by the Bank, we dealed with the fraud modalities *Lost/Stolen*, *Skimming*, *Mail Order*, *Account Take Over* and *Telephone Order*; and we did not manage other types like *Never Received Issuance*, *Manual Counterfeit* and *Fraud Application*.

[3] MCC got 33 categories so it could fit the number of groups of Transaction Category Code (TCC).

Table 1. Number of categories for each variable. *Previous* represents the value of the last transaction made for the same client.

name	mcc	mcc_previous	zip_code	zip_code_previous	value_trans
# of categ.	33	33	10	10	10
name	value_trans_previous	pos_entry_mode	credit_limit	brand	variant
# of categ.	10	10	10	6	6
name	score	type_person	type_of_trans	# of statements	speed
# of categ.	10	2	2	4	8
name	diff_score	credit_line	flag_fraud		
# of categ.	6	9	2		

Table 2. Number of frauds and legitimates in each split

base	1	2	3	4	5	6	7	8	9
development frauds	1,084	1,092	1,088	1,075	1,081	1,116	1,099	1,106	1,100
development legitimates	27,904	28,012	28,061	28,145	28,045	27,973	28,113	27,884	28,188
validation frauds	475	467	471	484	478	443	460	453	459
validation legitimates	12,184	12,076	12,027	11,943	12,043	12,115	11,975	12,204	11,960

Performance measures: In order to evaluate the classifiers, we have considered the use of KS, ROC Curve, Lift Curve, Precision (Hit Rate) and Recall accuracy (Detection Rate). From conversations with fraud prevention specialists and the first results using ROC curve and Hit Rate, we found out that we would obtain more appliable results if we used a cost function in which we adopted an average cost of $ 1 for every verification, and an average loss of $ 100 for every undetected fraud. This cost function combines Hit Rate and Detection Rate in one unique measure, and evaluates the function in only one point, the applicable cut-off. This was considered to be more similar to the used practice of a fraud score than a ROC curve that compares multiple references simultaneously. If we denote *tp*, *fp* and *fn* as the number of true positives (true frauds), false positive and false negatives, the final cost is given by:

$$\$cost = \$100 \times fn + \$1 \times (fp + tp).$$

Since the received database had only 10% of legitimate and 100% of fraudulent transactions, we had to adjust the cost function to:

$$\$cost = \$100 \times fn + \$10 \times fp + \$1 \times tp.$$

Once we prepared the data, we chose the methods to compare with the optimization criteria.

2 Parameter Space

In this small section we just introduce a very short description of the input parameters for the five chosen methods. A better description of these parameters can be found in the Appendix, and details about the methodologies and their

parameters can be found in Weka documentations [19,18] as well. The methods and their respective parameters are:

- NB has no parameter;
- DT has 2 parameters (C, M);
- BN has 3 parameters (D, Q, E) and 3 sub parameter (P, S, A);
- NN has 7 parameters (L, M, N, V, S, E, H);
- AIS has 9 parameters (S, F, C, H, R, V, A, E, K).

The methods NB and DT have a small parameter space. The parameter space of BN is also quite small, especially if we notice that there are few choices for many of them.

3 Optimization of Parameters

The parameter spaces of the methods Decision Tree, Bayesian Network and Naive Bayes are small enough in such a way that an exhaustive exploration of all possible parameter is possible. However, this is not the case for Neural Networks and Artificial Immune Systems. In order to find an optimized parameter set for these methods, we performed a parameters set optimization based on a Genetic Algorithm (GA).

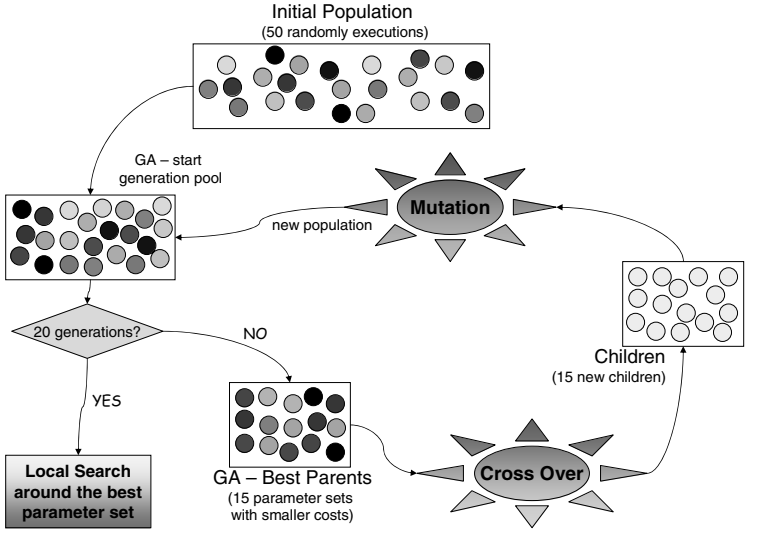


Fig. 1. Genetic Algorithm for parameters optimization

As showed in Figure 1, we start with an initial pool of 50 random executions, followed by 20 Genetic Algorithm (GA) generations. Each GA generation combines two randomly selected candidates among the best 15 from previous generation. This combination performs: cross over, mutation, random change

or no action for each parameter independently. As the generation goes by, the chance of no action increases. In the end, we perform a local search around the optimized founded by GA optimization. Notice that the final solution cannot be claimed to be optimal, and it is usually not optimal, but only suboptimal.

4 Robustness of the Parameters

Given a classification method M, after the parameter optimization, all optimized parameters may be independent of the split. In this case we say that this parameter set is *robust* and we name it $ROBUST(M)$.

When this does not happen, the optimization process is not as strong since the obtained optimized parameter set loses generalization power. In this case we decided to sacrifice prediction in order to gain robustness in the parameter set. In order to rewrite the optimization function that should be used in a GA algorithm, we have used a visualization procedure with computed costs for many equally spaced parameter sets in the parameter space. After defined a good optimization function, we proceeded not with another GA optimization because our time constraints, but we reused our initial runs used in the visualization, with the following kind of *multiresolution optimization* [9]:

1. we identify those parameters that have not changed, and we freeze these values for these respective parameters;
2. for any other parameter we screen the 20 best parameter sets for each split and identify reasonable range;
3. for all non-robust parameters, we choose an integer step s so the the searching space does not explode;
4. we evaluate the costs for all possible combinations according to the searching space defined above, and find the parameter set P that brings the minimum average cost among all the different used splits;
5. we zoom the screen to the neighborhood of P, refine steps s, and repeat the process from then on, until no refinement is possible.

In this case, after this process, we also call this parameters set *robust* and we name it $ROBUST(M)$. We should notice that we could also have used a GA optimization instead of a multiresolution optimization like the one performed by our multiresolution optimization.

In order to run the multiresolution optimization, we elected 6 splits (2,3,4,5,6 and 7) as the *robustization split group*, and 3 others (8,9 and 1) as the *evaluation split group* for posterior evaluation and comparison of all methods.

5 Results

We compare the following five classification methods: Naive Bayes (NB), Neural Network (NN), Bayesian Network (BN), Artificial Immune System (AIS) and Decision Tree(DT). For any method M, we have applied three different strategies:

$DEFAULT(M)$, $OPTIMIZED(M)$ and $ROBUST(M)$, in which $DEFAULT$ means to use default parameters provided by Weka; $OPTIMIZED$ refers to an optimized set of parameters obtained as described in Section 3, and $ROBUST$ is an optimized robust set of parameters.

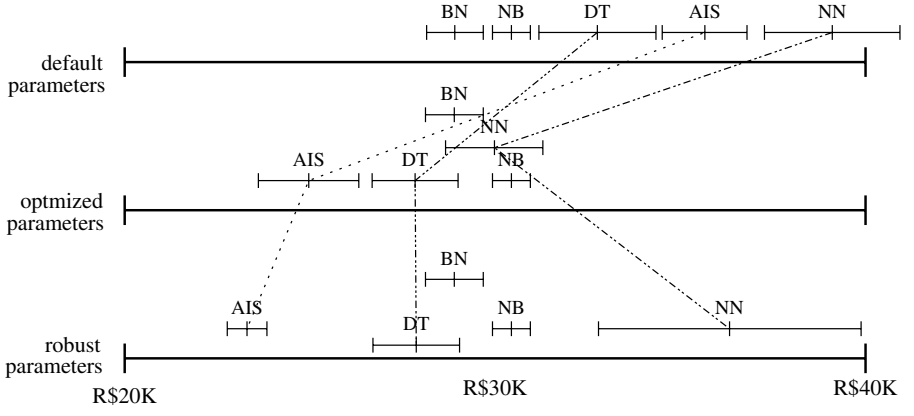


Fig. 2. Summary results for the methods in all strategies. Average and standard deviation (statistics based on the 3 evaluation splits) are represented by small error-bars, for the 5 methods, for the 3 strategies. The figure is divided in three stacked horizontal lines with their methods statistics (the error-bars) in order to separate strategies: default parameters, optimized parameters and robust parameters, in order of evolution. All 3 large horizontal lines represent the cost functions, ranging from R$ 20 thousand in the left end to R$ 40 thousand in the right end. In order to better display the error-bars, some of them were vertically shifted. AIS led to the smallest cost with robust parameters, followed by DT, and NN led to the largest cost.

Table 3. Summary results for the methods in all strategies. Average and standard deviation for the 3 evaluation splits.

Strategy	DT	AIS	BN	NN	NB
DEFAULT	32.76 (4.83%)	35.66 (3.21%)	**28.91** (2.65%)	39.10 (4.68%)	30.44 (1.68%)
OPTIMIZED	27.84 (4.16%)	**24.97** (5.43%)	28.90 (2.69%)	29.98 (4.38%)	30.44 (1.68%)
ROBUST	27.87 (4.21%)	**23.30** (2.29%)	28.90 (2.69%)	36.33 (9.75%)	30.44 (1.68%)

One can see in Figure 2 and Table 3 the final costs of the classification methods obtained for all strategies. We show here only the average costs with their standard deviations for the 3 splits used for evaluation of the robust parameter sets. The cost is represented in thousand of Reais (Brazilian Currency), the smaller, the better. The standard deviations (num%) are considered in the same way as errors. From these results one can notice that:

– The Bayesian methods BN and NB are such that their results are independent from the used strategies. This is expected for NB, since there are no parameters. For BN, the default parameters performed almost in the same

way as the optimized strategies, independently from the splits. The maximum number of node parents influences the final topology and probability tables but not enough to impact the final costs;
- For strategy $DEFAULT$ we used the default parameters. BN was the best method. AIS and NN got relatively poor results compared to the others. Particularly, NN improved only 15.4%[4] in relation to a strategy which considers all transactions as legitimate;
- For what concerns the strategy $OPTIMIZED$ with optimized parameters, we verified that almost all the methods led to reduced costs in comparison to the case with default parameters. The method that reduced its cost the most, with 29.98%[5] of cost reduction, was AIS and it became the best method for this strategy. The second best method was DT, that reached a 15.01% of cost reduction. NN reduced its cost by 23.33%[6] ;
- When we analyzed the strategy $ROBUST$, we saw two important facts: first, there was an abrupt cost increase for $ROBUST(NN)$ in relation to $OPTIMIZED(NN)$, that shows the over-fitting tendency of method NN with optimized parameters. There was a cost reduction for $ROBUST(AIS)$ in relation to $OPTIMIZED(AIS)$. We suppose that this happened due to the fact that AIS has more parameters and also the largest parametric search space. In this way, when the parametric space is reduced, after the freezing of some parameters during the parameters robustization process, it can be observed a more efficient optimization. This phenomenon is many times mentioned as "Curse of Dimensionality".

Robust set of parameters: The table 4 shows the set of optimized robust parameters for each method.

At first glance, we can observe that for DT we have a tree with minimum pruning according to parameter M. For NN, we see that the parameters L and M achieved very interesting values with a big L (Learning Rate) and very small M (Momentum). This fact allows us to trace a parallel with DT, saying that, as well as DT, NN takes a step to less pruning and more over-fitting. BN was already optimal with default parameters. Finally, for AIS, we obtained a very good set of parameters from GA execution, which made the multiresolution optimization phase quite easy in order to obtain a good optimized and robust set of parameters. One of the most surprising results was K equals to 1, which means that no voting is necessary: the first rule that matches decides the class.

Final comparison of all methods: Since the standard deviation seen in Figure 2 suggests us that DT, BN and NB could have the same costs, we performed four statistics t-student tests with 100 new random splits in the same proportion.

[4] 15.4% = \$39.1 thousands/\$46.2 thousands, where \$46.2 thousands corresponds to the average cost of the validation part of the splits 8, 9 and 1 when one simply decides letting frauds happen unwatched.

[5] 29.98% = 1 - \$ 24.97 thousands / \$ 35.66 thousands = $1 - OPTIMIZED(AIS)/DEFAULT(AIS)$.

[6] 23.33% = 1 - \$ 29.98 thousands / \$ 39.10 thousands = $1 - OPTIMIZED(NN)/DEFAULT(NN)$.

Table 4. Summary of optimized robust parameters. Parameters N,S for NN and A,S for AIS were not iterated. Parameters E,V for NN and K,F,H,V for AIS were frozen for the multiresolution optimization. Parameters L,M,H for NN and C,R,E for AIS needed a multiresolution optimization. Parameter H=20 in NN is the number of attributes + number of classes + 1, parameter P=17 for BN is the number of attributes.

Method	Average Cost on validation	Robust parameters in command line display
DT	$ 27,870.66	-C 0.49 -M 1
NB	$ 30,439.33	n/a
BN	$ 28,901.66	-D -Q weka.classifiers.bayes.net.search.local.K2 – -P 17 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator – -A 0.5
NN	$ 36,332.33	-L 0.40 -M 0.12 -H 20 -E 0 -V 0 -N 500 -S 0
AIS	$ 23,303.00	-C 30 -R 177 -E 5 -K 1 -F 0 -H 10 -V 1 -A -1 -S 1

These splits were specially created for these tests. We tested if $ROBUST(AIS) - ROBUST(DT) = 0$, $ROBUST(DT) - ROBUST(BN) = 0$, $ROBUST(BN) - ROBUST(NB) = 0$ and $ROBUST(NB) - ROBUST(NN) = 0$. Not surprisingly, with 99.9% of certainty, all $H0$ were rejected, which means that none of them is equal. In the end, the average of costs for strategy robust is what defines the rank of methods. From the Figure 2, we can notice that AIS produced the best classifiers, followed by DT, BN, NB, and NN, in this order.

6 Future Work

We intend to analyze in details the optimized parameters in the coming future, and try to reach better relations between the value of each parameter and its relation to the skewness of the data, at same time that we enquire why AIRS2 implementation of AIS outperforms the implementations of other methods. We are also extending the analysis in such a way to evaluate the influence of a metaheuristics like Cost Sensitive Classifier [5], which takes into account the different costs of false positive and false negative in the training phase. Using this metaheuristics, in our preliminary and unfinished results, we are observing that one may obtain better classifiers for all methods, up to Naive Bayes. We also consider the inclusion of Support Vector Machines (SVM) in the pool of compared methods. And given we are using AIS, a suitable comparison method would be k nearest neighbour.

We intend to apply the models for unseen out-of-date datasets to compare stability and life expectancies. Since, as we know, the fraudulent behavior is very dynamic, often a model loses its prediction power in a short time. Besides knowing which method generates the most accurate model, it is important to know which one generates the model that remains predictive for a longer time.

7 Conclusions

In this paper, we present a comparative study of five classification methods (Decision Tree, Neural Network, Bayesian Network, Naive Bayes and Artificial

Immune System). The used definition of an objective function to be optimized that takes into account different costs for false positives and false negatives is important. In all our executions, except for NB (no parameter needed) and BN, we concluded that the best results had not been reached with default set of parameters as given in Weka. Particularly for AIS and NN, the results gotten using default parameters are very poor if compared with those gotten after a parametric adjustment using GA. Our tests results show that BN is better than NN, the most used method in real application today, which reproduces the results from Maes [11,12]. In addition, we obtained that AIS and DT also surpass NN. Perhaps because DT is a classic classification method, it has been forgotten in recent works. However, it still reveals itself as one of the best methods, with sufficient competitive results. On our tests AIS had a surprisingly large increase of performance from default parameters to GA optimized parameters, and this performance was kept in the obtaining of an optimized robust parameter set.

To sum up, AIS produced the best classifiers, followed by DT, BN, NB, and NN, respectively.

Acknowledgments

This work was partially supported by Projects PRONEX-FAPESP # 03/09925-5 and CNPq # 306217/2004-0. We would like to acknowledge the banks Santander and Citybank for supporting the development of this reasearch by their employees. We would also like to thank Roberto Cesar Marcondes, Nina Hirata, Casimir Kulikowski, João Eduardo Ferreira, Yoshiharu Kohayakawa for their comments and support.

References

1. Brownlee, J.: Artificial immune recognition system (airs) - a review and analysis. Technical report, Victoria, Australia: Centre for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology (January 2005)
2. Charniak, E.: Bayesians networks without tears. AI Magazine (1991)
3. DasGupta, D.: Artficial Immune Systems and Their Applications. Springer, New York (1998)
4. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A Novel Paradigm to Pattern Recognition. University of Paisley (2002)
5. Elkan, C.: The foundations of cost-sensitive learning. In: IJCAI, pp. 973–978 (2001)
6. Fan, W., Stolfo, S.J., Zhang, J., Chan, P.K.: AdaCost: misclassification cost-sensitive boosting. In: Proc. 16th International Conf. on Machine Learning, pp. 97–105. Morgan Kaufmann, San Francisco (1999)
7. Holland, J.: Adaptation in Natural and Artificial Systems, 1st edn. MIT press, Cambridge (1975)
8. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Science, vol. 79, pp. 2554–2558 (1982)

9. Kim, J., Zeigler, B.P.: A framework for multiresolution optimization in a parallel/distributed environment: simulation of hierarchical gas. J. Parallel Distrib. Comput. 32(1), 90–102 (1996)
10. Klarreich, E.: Inspired by immunity. Nature (415), 468–470 (2002)
11. Maes, S., Tuyls, K., Vanschoenwinkel, B.: Machine learning techniques for fraud detection (2000)
12. Maes, S., Tuyls, K., Vanschoenwinkel, B., Manderick, B.: Credit card fraud detection using bayesian and neural networks. In: Proceedings of NF 2002, Havana, Cuba, January 16-19 (2002)
13. Murthy, S.K., Ohlebusch, E., Kurtz, S.: Automatic construction of decision trees from data: A multi-disciplinary survey. In: Data Mining and Knowledge Discovery, USA, vol. 2, pp. 345–389. Kluwer Academic Publishers, Dordrecht (1998)
14. Phua, C., Lee, V., Smith, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research. Artificial Intelligence Review (submitted for publication, 2005)
15. Schapire, R.E., Singer, Y.: Improved boosting using confidence-rated predictions. Machine Learning 37(3), 297–336 (1999)
16. Stolfo, S., Fan, W., Lee, W., Prodromidis, A., Chan, P.: Credit card fraud detection using meta-learning: Issues and initial results, 1997. In: Working notes of AAAI Workshop on AI Approaches to Fraud Detection and Risk Management (1997)
17. Watkins, A., Timmis, J., Boggess, L.: Artificial immune recognition system (AIRS): An immune-inspired supervised machine learning algorithm. Genetic Programming and Evolvable Machines 5(3), 291–317 (2004)
18. Witten, I.H., Franku, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Elsevier, Amsterdam (2005)
19. Witten, I.H., Franku, E.: Software documentation: Weka (2008), `Weka-3-4-11.doc/weka/classifiers/functions/MultilayerPerceptron.html`

Appendix

For next paragraph, let us define VR = [X1;X2; step = S] as been the allowed variation range from X1 to X2 and S, the precision step for this specific parameter S.

Naive Bayes: **NB** does not have any parameter.

Decision Tree: **DT** has two parameters C and M:

- **C:** the confidence threshold for pruning. (Default: 0.25). VR = [0.01;1.00; step = 0.01].
- **M:** the minimum number of instances per leaf. (Default: 2). VR = [1;100; step = 1].

Bayesian Network: **BN** has three parameters (D, Q, E):

- **D:** defines whether a structure called ADTree will or not be used;
- **Q:** defines which search for topology algorithm will be used. The available ones are: GeneticSearch, HillClimber, K2, LocalScoreSearchAlgorithm, RepeatedHillClimber, SimulatedAnnealing, TabuSearch e TAN. Every search algorithm has two parameters:

  - **P:** defines the number of parent's allowed in the topology.
  - **S:** defines the type of score to be used to build the conditional table, they are: BAYES, BDeu, MDL, ENTROPY e AIC;
- **E:** defines the estimator algorithm to calculate the conditional tables. In Weka they are: BayesNetEstimator, BMAEstimator, MultiNomialBMAEstimator and SimpleEstimator (this estimator has one parameter (A), called alpha, and it ranges between 0% e 100%, and it represents a start value for the conditional probability.).

Neural Network: **NN** has seven parameters (L, M, N, V, S, E, H):

- **L:** the learning rate. (default 0.3). The closer to zero, the smaller the impact of the incoming information to be learnt. VR = [0.01;1.00; step = 0.01].
- **M:** the momentum (default 0.2). Its inclusion (values greater than zero) has for objective to increase the speed of the training of a neural net and to reduce the instability. VR = [0.00;1.00; step = 0.01].
- **N:** the number of epochs to train through. (default 500). our tests indicates that using N greater than 500 does not increase the performance significantly, and fixing it to its default 500. VR = [500;500; step = 0].
- **V:** the percentage size of the validation set from the training to use. (default 0 (no validation set is used, instead number of epochs is used). It ranges between 0% and 99,99%, when this parameter is greater that zero intend to reduce over-fitting. VR = [0.00;0.99; step = 0.01].
- **S:** the seed for the random number generator. We used default value. VR = [0;0; step = 0].
- **E:** the threshold for the number of consecutive errors allowed during validation testing. (default 20). Number between 1 and 100. This parameter participates with N to form the stop condition of the algorithm. VR = [1;100; step = 1].
- **H:** string of numbers of nodes to be used on each layer. Each number represents its own layer and the number of nodes on that layer. There are also some wildcards: 'a', 'i', 'o', 't'. These are 'a' = (number of attributes + number of classes) / 2, 'i' = number of attributes, 'o' = number of classes, and 't' = number of attributes + number of classes. VR = [1;20; step = 1].

Artificial Immune System: **AIS** has 9 parameters (S, F, C, H, R, V, A, E, K):

- **S:** the seed for the random number generator. (default 0). We adopted the fixed value 1. VR = [1;1; step = 0].
- **F:** the minimum number percentage affinity threshold (see [17] page 6). VR = [0.00;0.5; step = 0.01].
- **C:** the Clonal Rate is an integer that ranges between 0 ant 100. VR = [1;100; step = 1].
- **H:** the Hyper-mutation rate. Ranges between 0 and 100 and determines the percentage of clones (from last parameter) that will suffer mutation. VR = [0;10; step = 1].

- **R:** the total resources is the maximum number of B-Cell (or ARB) allowed in the system. VR = [0;200; step = 1].
- **V:** the Stimulation threshold is a number between 0 and 1 used as criteria to keep or drop a given B-Cell. VR = [0.00;1.00; step = 0.01].
- **A:** the number of affinity threshold instances. Because of lack of documentation in [1] we used the default (-1) value. VR = [-1;-1; step = 0].
- **E:** the memory pool size. Define the number of random initialization instances. By simplicity we varied it between 0 and 10. VR = [0;10; step = 1].
- **K:** the number of nearest neighbors representing B-Cells to be matched and consulted in a voting election of which class the current transaction belongs to. K equals to 1 means no voting. VR = [0;10; step = 1].

Artificial Immune Recognition System with Nonlinear Resource Allocation Method and Application to Traditional Malay Music Genre Classification

Shahram Golzari[1,2], Shyamala Doraisamy [1], Md Nasir B. Sulaiman [1], Nur Izura Udzir [1], and Noris Mohd. Norowi [1]

[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400, Serdang, Selangor, Malaysia
[2] Electrical and Computer Engineering Department, Hormozgan University, Bandarabbas, Iran
golzarihormozi@yahoo.com,
{shyamala,nasir,izura,noris}@fsktm.upm.edu.my

Abstract. Artificial Immune Recognition System (AIRS) has shown an effective performance on several machine learning problems. In this study, the resource allocation method of AIRS was changed with a nonlinear method. This new algorithm, AIRS with nonlinear resource allocation method, was used as a classifier in Traditional Malay Music (TMM) genre classification. Music genre classification has a great important role in music information retrieval systems nowadays. The proposed system consists of three stages: feature extraction, feature selection and finally using proposed algorithm as a classifier. Based on results of conducted experiments, the obtained classification accuracy of proposed system is 88.6 % using 10 fold cross validation for TMM genre classification. The results also show that AIRS with nonlinear allocation method obtains maximum classification accuracy for TMM genre classification.

Keywords: Artificial Immune System, AIRS, Music Genre Classification, Nonlinear Resource allocation.

1 Introduction

Interest on music information retrieval systems for the storage, retrieval and classification of large collections of digital musical files has grown in recent years. Metadata such as filename, author, file size, date and genres are commonly used to classify and retrieve these documents. Such manual classification is highly labor-intensive and costly both in terms of time and money [1]. An automatic classification system that is able to analyze and extract implicit knowledge of the musical files is therefore highly sought. One approach to automated musical classification that is currently being widely studied is classification based on musical genres.

Musical genres are labels created and used by humans for categorizing and describing music [2]. Examples of a few Western musical genres are such as Pop, Rock, Hip-hop, and Classical. Several systems for automated genre classification and retrieval

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 132–141, 2008.

of musical files have been researched and developed [2], [3]. However, most of these studies were conducted using only western dataset and we focus on non-Western musical genres, and more specifically on Traditional Malay Music (TMM). Norowi *et al* [4] have shown the significance effect of beat features for TMM genre classification in comparison to Western musical genres. Therefore the behavior of TMM genres is different from western music genres and need more studies to explore it.

Artificial immune system (AIS) is a computational method inspired by the biology immune system. It is progressing slowly and steadily as a new branch of computational intelligence and soft computing [5],[6]. One of AIS based algorithms is Artificial Immune Recognition System (AIRS). AIRS is a supervised immune-inspired classification system capable of assigning data items unseen during training to one of any number of classes based on previous training experience. AIRS is probably the first and best known AIS for classification, having been developed in 2001 [7].

In this study, the proposed approach consists of three stages: feature extraction, feature selection preprocessing and finally classification with AIRS. Feature selection is used to improve the quality of data that has been extracted in manual manner. Also non linear resource allocation is used in AIRS to increase its classification performance by means of resource number. The performance of the proposed method was tested with regard to classification accuracy. The obtained classification accuracy of our method is 88.6% using ten fold cross validation for TMM genre classification. Based on the results, AIRS with nonlinear resource allocation has most accuracy among the classifiers that used in the experiments.

The remainder of this paper is organized as follows: Section 2 gives the briefly description about Traditional Malay Music. Section 3 and 4 describe about feature extraction and feature selection methods, respectively. AIRS and using nonlinear resource allocation method in AIRS are explained in Section 5. In Section 6, we explain the experiments and discuss about the results and consequently in Section 7, we conclude the paper.

2 Traditional Malay Music

Traditional Malay music is mainly derivative, influenced by the initial overall Indian and Middle Eastern music during the trade era and later from colonial powers such as Thailand, Indonesia, Portuguese and British who introduced their own culture including dance and music. A thorough overview on the origin and history of TMM can be found in [8]. The taxonomy of TMM depends on the nature of the theatre forms they serve and their instrumentations. Categorization of TMM genres has been studied extensively by Ang [9]. Music of these genres is usually disseminated non-commercially, usually performed by persons who are not highly trained musical specialists, undergoes change arising from creative impulses and exists in many forms. The musical ensembles usually include gendangs or drums that are used to provide constant rhythmic beat of the songs and gongs to mark the end of a temporal cycle at specific part of the song [10].

One common attribute that is shared by most TMM genres is that they are generally repetitive in nature and exist in ‘gongan’-like cycle. ‘Gongan’ is defined as a temporal cycle marked internally at specific points by specific gongs and at the end

by the lowest-pitched gong of an ensemble [8]. It is an important structural function as it divides the musical pieces into temporal sections. Once every measure has been played, musicians continue playing in a looping motion by repeating the cycle from the beginning again until one of the lead percussionists signals the end of the song by varying their rhythms noticeably. In general, TMM does not have a chorus that plays differently than other parts of the songs, which is the usual occurrence in western music. Its repetitiveness and constant rhythms are two aspects that are taken into account to facilitate classification by genre later.

Very little study has been conducted on automatic traditional Malay music genre classification in the literature. Norowi *et al* [4] studied the effects of various factors and audio feature set combinations towards the classification of TMM genres. Results from experiments conducted in several phases show that factors such as dataset size, track length and location¸ together with various combinations of audio feature sets comprising Short Time Fourier Transform (STFT), Mel-Frequency Cepstral Coefficients (MFCCs) and Beat Features affect classification. This study also only used the J48 classifier and achieved 66.3% classification accuracy for TMM genres [4]. We could not find more researches about TMM genre classification. In this study, we propose the hybrid system that includes feature extraction, feature selection and AIRS classifier with new resource allocation method to improve the performance of automatic TMM genre classification.

3 Feature Extraction

Ten TMM genres were involved in this study. The breakdown for each genre and its number of musical files are listed in Table 1. A relatively small dataset was used in this experiment due to the difficulty in obtaining digital files of TMM, as traditional Malay musical culture is fast corroding with little preservation in digital format. Whilst it was much easier to obtain dataset for western music, the number was also kept small to match the size of TMM dataset.

Musical files for this experiment were obtained from the Malaysia Arts Academy, Sultan Salahuddin Abdul Aziz Shah's Cultural and Arts Centre at Universiti Putra Malaysia, Student's Cultural Centre at Universiti Malaya and also personal collections of audio CDs from many individuals. The dataset became available in both digital and analog format. Quite a number of musical data for TMM genres were in analog format and were digitized manually. All of the digital music files were then converted into wav files; the only audio format supported by the existing feature extraction tool used at the time of study. The whole dataset was later trimmed to specific length and location in the file by executing certain audio commands through batch processing before extraction began.

The features were extracted from the music files through MARSYAS-0.2.2; a free framework that enables the evaluation of computer audition applications. MARSYAS is a semi-automatic music classification system that is developed as an alternative solution for the existing audio tools that are incapable of handling the increasing amount of computer data [2]. It enables the three feature sets for representing the timbral texture, rhythmic content and pitch content of the music signals and uses trained statistical pattern recognition classifiers for evaluation.

Table 1. Overall number of musical files for each genre

NO	Genre	Class Label	Number
1	Dikir Barat	A	31
2	Etnik Sabah	B	12
3	Gamelan	C	23
4	Ghazal	D	17
5	Inang	E	10
6	Joget	F	15
7	Keroncong	G	43
8	Tumbuk Kalang	H	13
9	Wayang Kulit	I	17
10	Zapin	J	10

4 Feature Selection

Feature selection is the process of removing features from the data set that are irrelevant with respect to the task that is to be performed. Feature selection can be extremely useful in reducing the dimensionality of the data to be processed by the classifier, reducing execution time and improving predictive accuracy (inclusion of irrelevant features can introduce noise into the data, thus obscuring relevant features). It is worth noting that even though some machine learning algorithms perform some degree of feature selection themselves (such as classification trees), feature space reduction can be useful even for these algorithms. Reducing the dimensionality of the data reduces the size of the hypothesis space and thus results in faster execution time.

Feature selection techniques can be split into two categories – filter methods and wrapper methods. Filter methods determine whether features are predictive using heuristics based on characteristics of the data. Wrapper methods make use of the classification algorithm that will ultimately be applied to the data in order to evaluate the predictive power of features. Wrapper methods generally result in better performance than filter methods because the feature selection process is optimized for the classification algorithm to be used. However, they are generally far too expensive to be used if the number of features is large because each feature set considered must be evaluated with the trained classifier. For this reason, wrapper methods will not be considered in this study. Filter methods are much faster than wrapper methods and therefore are better suited to high dimensional data sets. We have used Gain Ratio (GR) feature evaluation method. Since the GR does not perform feature selection but only feature ranking, this method usually is combined with searching strategy in feature subset space when one needs to find out the appropriate number of features. Forward selection, backward elimination, bi-directional search, best-first search, genetic search, and other methods are often used on this task. Specifically, we experimented with the best first search in this study. For detailed information about GR, readers are referred to [11], [12].

5 AIRS

Artificial Immune Recognition System (AIRS) is investigated by Watkins [7]. AIRS can be applied to classification problems, which is a very common real world data mining task. Most other artificial immune system research concerns unsupervised learning and clustering. The only other attempt to use immune systems for supervised learning is the work of Carter [13]. The AIRS design refers to many immune system metaphors including resource competition, clonal selection, affinity maturation, memory cell retention and also used the resource limited artificial immune system concept investigated by [14]. In this algorithm, the feature vectors presented for training and test are named as antigens while the system units are called as B cells. Similar B cells are represented with Artificial Recognition Balls (ARBs) and these ARBs compete with each other for a fixed resource number. This provides ARBs, which have higher affinities to the training antigen to improve. The memory cells formed after the whole training antigens were presented are used to classify test antigens.

AIRS has four stages. The first is performed once at the beginning of the process (normalization and initialization), and other stages constitute a loop and are performed for each antigen in the training set: ARB generation, Competition for resources and nomination of candidate memory cell, promotion of candidate memory cell into memory pool. The mechanism to develop a candidate memory cell is as follows [7], [15]:

1. A training antigen is presented to all the memory cells belonging to the same class as the antigen. The memory cell most stimulated by the antigen is cloned. The memory cell and all the just generated clones are put into the ARB pool. The number of clones generated depends on the affinity between the memory cell and antigen, and affinity in turn is determined by Euclidean distance between the feature vectors of the memory cell and the training antigen. The smaller the Euclidean distance, the higher the affinity, the more is the number of clones allowed.
2. Next, the training antigen is presented to all the ARBs in the ARB pool. All the ARBs are appropriately rewarded based on affinity between the ARB and the antigen as follows: An ARB of the same class as the antigen is rewarded highly for high affinity with the antigen. On the other hand, an out of class ARB is rewarded highly for a low value of affinity measure. The rewards are in the form of number of resources. After all the ARBs have been rewarded, the sum of all the resources in the system typically exceeds the maximum number allowed for the system. The excess number of resources held by ARBs are removed in order starting from the ARB of lowest affinity and moving higher until the number of resources held does not exceed the number of resources allowed for the system. Those ARBs, which are not left with any resources, are removed from the ARB pool. The remaining ARBs are tested for their affinities towards the training antigen. If for any class of ARB the total affinity over all instances of that class does not meet a user defined stimulation threshold, then the ARBs of that class are mutated and their clones are placed back in the ARB pool. Step 2 is repeated until the affinity for all classes meet the stimulation threshold.

3. After ARBs of all classes have met the stimulation threshold, the best ARB of the same class as the antigen is chosen as a candidate memory cell. If its affinity for the training antigen is greater than that of the original memory cell selected for cloning at step 1, then the candidate memory cell is placed in the memory cell pool. If in addition to this the difference in affinity of these two memory cells is smaller than a user defined threshold, the original memory cell is removed from the pool.

These steps are repeated for each training antigen. After completion of training the test data are presented only to the memory cell pool, which is responsible for actual classification. The class of a test antigen is determined by majority voting among the k most stimulated memory cells, where k is a user defined parameter.

Some researches have been done to evaluate the performance of AIRS [15], [16], [17], [18], [19]. The results show that AIRS is comparable with famous and powerful classifiers.

5.1 Nonlinear Resource Allocation

Resource competition is one stage of AIRS. The purpose of resource competition stage is improving the selection probability of high-affinity ARBs for next steps. Resource competition is done based on the number of allocated resources for each ARB. According to this resource allocation mechanism, half of resources is allocated to the ARBs in the class of Antigen while the remaining half is distributed to the other classes. The distribution of resources is done by multiplying stimulation rate with clonal rate that shown in (1). Mervah *et al* [15] have used a different resource allocation mechanism. In their mechanism, the Ag classes occurring more frequently get more resources. Classical AIRS and Mervah study use the linear resource allocation and the number of allocated resources has linearly relation with affinities. In linearity approach the difference in allocated resource number between high affinity ARBs and low affinity ARBs is not very wide. Therefore, the more number of low affinity ARBs remain in the system and algorithm uses excess resources.

$$Rsources = StimulationRate \times ClonalRate \tag{1}$$

In this study, we use the nonlinear coefficient for clonal rate in (1) to solve this problem. The appropriate nonlinear coefficient should allocate more resources for high affinity ARBs and less resources for low affinity ARBs in comparison to linear method. Resource allocation is done in nonlinearly with affinities, by using this type of coefficient. Also, the difference in resources number between high-affinity ARBs and low affinity ARBs is bigger in this approach than linear approach. In this study, we use very simple mathematic function to satisfy maintained condition. This function is shown in (2). To evaluate the proposed method, we apply the AIRS with this nonlinear resource allocation method to TMM genre classification and compare its accuracy to accuracies of some famous classifiers.

$$Rsources = \begin{cases} (Stimulatio\ n\ Rate)^{\frac{1}{2}} \times ClonalRate & if\ \ Stimulatio\ n\ Rate \geq 0.5 \\ (Stimulatio\ n\ Rate)^{2} \times ClonalRate & if\ \ Stimulatio\ n\ Rate < 0.5 \end{cases} \tag{2}$$

6 Experiments and Results

In this study, the feature extraction method, described in section 3, was used to extract the TMM features. The result of this phase was a data set with 63 features and 193 instances. After that, we used the GR feature subset evaluation with best first search strategy to reduce the dimensional of data set. The feature selection method reduced the number of features to 25 features.

Some experiments were carried out in order to determine how AIRS with nonlinear resource allocation method performed TMM genre classification in compared to AIRS and some other famous classifiers. One advantage of AIRS is that it is not necessary to know the appropriate settings and parameters for the classifier. The most important element of the classifier is its ability to be self-determined. The used values of AIRS parameters can be found in Table 2.

Table 2. Algorithm Parameters

Used Parameter	Value
Clonal rate	10
Mutation rate	0.1
ATS	0.2
Stimulation threshold	0.99
Resources	150
Hyper mutation rate	2.00
K value in KNN classifier	4

As we mentioned earlier, we couldn't find more researches about TMM genre classification problem. Therefore, to evaluate the performance of proposed method, the follow classifiers were chosen.

- Bagging
- Bayesian Network
- Cart
- Conjunctive rule learner (Conj-Rules)
- Decision Stump
- Decision Table
- IB1
- J48 (an implementation of C4.5)
- Kstar
- Logistic
- LogitBoost
- Multi-layer neural network with back propagation (MLP)
- Naïve Bayesian
- Nbtree
- PART (a decision list [20])
- RBF Network
- SMO (a support vector machine [21])

Table 3. TMM Genre Classification Accuracies

Method	Accuracy (%)
Conj-Rules	31.60
Decision Stump	33.68
Decision Table	52.85
CART	61.67
PART	68.39
J48	73.06
Nbtree	75.13
Bagging	76.68
Naïve Bayesian	77.72
RBF	80.31
Bayesian Network	80.83
Kstar	80.83
LogitBoost	81.35
MLP	84.47
IB1	84.97
Logistic	86.01
SMO	86.01
AIRS	86.01
The Proposed Method	88.60

Table 4. Confusion Matrix

	A	B	C	D	E	F	G	H	I	J
A	29	0	0	0	0	2	0	0	0	0
B	2	6	0	0	0	0	0	0	4	0
C	0	0	22	0	1	0	0	0	0	0
D	0	0	0	17	0	0	0	0	0	0
E	0	0	0	0	8	0	0	0	0	2
F	0	0	1	0	2	12	0	0	0	1
G	0	0	0	0	0	0	41	1	0	0
H	0	0	0	0	1	0	0	12	0	0
I	0	0	0	0	1	0	2	0	14	0
J	0	0	0	0	0	0	0	0	0	10

This list includes a wide range of paradigms. The code written on the WEKA [12] data mining package and the default parameters were used for each algorithm.

A 10-fold cross validation approach was used to estimate the predictive accuracy of the algorithms. In this approach, data instances are randomly assigned to one of 10 approximately equal size subsets. At each iteration, all but one of these sets are merged to form the training set while the classification accuracy of the algorithm is measured on the remaining subset. This process is repeated 10 times, choosing a different subset as the test set each time until all data instances have been used 9 times

for training and once for testing. The final predictive accuracy is computed over all folds in the usual manner but dividing the number of correct classifications taken over all folds by the number of data instances in all folds. This approach was used in all experiments to control the validity of experiments.

The achieved accuracies by classifiers are shown in Table 3. Based on the results, the proposed method not only increases the accuracy of AIRS from 86.1% to 88.6 %, but also has most accuracy among the classifier.

Table 4 shows the confusion matrix obtained by applying proposed method to TMM genre classification. Results show the class B has the worst behavior among classes and only 50% of this class instances are classified truly. More exploration on the data collection and feature extraction for this class can be done in feature work to achieve more accuracy.

7 Conclusions

AIRS is the most important classifier among the Artificial Immune System based classifiers. In this study, the resource allocation mechanism of AIRS was changed with a nonlinear resource allocation method. In the application phase of this study, this new version of AIRS was used to classify Traditional Malay Music genres. Some experiments were conducted to see the effects of proposed resource allocation method. According to experimental results, AIRS with nonlinear resource allocation method showed a considerably high performance with regard to the classification accuracy for Traditional Malay Music genres. The obtained classification accuracy of proposed algorithm for Traditional Malay Music genre classification was 88.6%. Also this accuracy was maximum accuracy among accuracies that obtained by used classifiers in experiments.

References

1. Dannenberg, R., Foote, J., Tzanetakis, G., Weare, C.: Panel: New Directions in Music Information Retrieval. In: International Computer Music Conference, International Computer Music Association, pp. 52–59 (2001)
2. Tzanetakis, G., Cook, P.: Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing 10(5) (2002)
3. Wold, E., Blum, T., Keislar, D., Wheaton, J.: Content-based Classification, Search, and Retrieval of Audio. IEEE Multimedia 3(3), 27–36 (1996)
4. Norowi, N.M., Doraisiamy, S., Rahmat, R.W.: Traditional malaysian musical genres classification based on the analysis of beat feature in audio. Journal of Information Technology in Asia, JITA 2 (2007)
5. de Castro, L.N., Timmis, J.: Artificial Immune Systems as a novel Soft Computing Paradigm. Soft Computing Journal 7(7) (2003)
6. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
7. Watkins, A.: AIRS: A Resource Limited Artificial Immune Classifier. M.S. thesis, Department of Computer Science. Mississippi State University (2001)

8. Matusky, P.: Malaysian Shadow Play and Music: Continuity of an Oral Tradition. Oxford University Press, Kuala Lumpur (1993)
9. Ang, M.: A Layered Architectural Model for Music: Malaysian Music on the World Wide Web. Ph.D. dissertation, UPM (1998)
10. Becker, J.: The Percussive Patterns in the Music of Mainland Southeast Asia. Ethnomusicology 2(2), 173–191 (1968)
11. Hall, M.A., Smith, L.A.: Practical feature subset selection for machine learning. In: Proceedings of the 21st Australian Computer Science Conference, pp. 181–191 (1998)
12. Witten, H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
13. Carter, J.H.: The immune systems as a model for pattern recognition and classification. Journal of the American Medical Informatics Association 7(1), 28–41 (2000)
14. Timmis, J., Neal, M.: A Resource Limited Artificial Immune System. Knowledge Based Systems 14(3), 121–130 (2001)
15. Marwah, G., Boggess, L.: Artificial immune systems for classification: Some issues. In: Proceedings of the first international conference on artificial immune systems, University of Kent Canterbury, England, pp. 149–153 (2002)
16. Watkins, A., Boggess, L.: A new classifier based on resource limited artificial immune systems. In: Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence, Honolulu, HI, pp. 1546–1551 (2002)
17. Watkins, A., Timmis, J.: Artificial Immune Recognition System (AIRS): Revisions and Refinements. In: 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK, pp. 173–181 (2002)
18. Watkins, A.: Exploiting Immunological Metaphors in the Development of Serial, Parallel, and Distributed Learning Algorithms. PhD Thesis, Computer Science, University of Kent, Canterbury, England (2005)
19. Watkins, A., Timmis, J., Boggess, L.: Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. Genetic Programming and Evolvable Machines 5(3), 291–317 (2004)
20. Frank, E., Witten, I.H.: Generating Accurate Rule Sets without Global Optimization. In: Fifteenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco (1998)
21. Keerthi, S.S., et al.: Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation 13(3), 637–649 (2001)

Further Exploration of the Dendritic Cell Algorithm: Antigen Multiplier and Time Windows

Feng Gu, Julie Greensmith, and Uwe Aickelin

School of Computer Science, University of Nottingham, UK
{fxg,jqg,uxa}@cs.nott.ac.uk

Abstract. As an immune-inspired algorithm, the Dendritic Cell Algorithm (DCA), produces promising performance in the field of anomaly detection. This paper presents the application of the DCA to a standard data set, the KDD 99 data set. The results of different implementation versions of the DCA, including antigen multiplier and moving time windows, are reported. The real-valued Negative Selection Algorithm (NSA) using constant-sized detectors and the C4.5 decision tree algorithm are used, to conduct a baseline comparison. The results suggest that the DCA is applicable to KDD 99 data set, and the antigen multiplier and moving time windows have the same effect on the DCA for this particular data set. The real-valued NSA with contant-sized detectors is not applicable to the data set. And the C4.5 decision tree algorithm provides a benchmark of the classification performance for this data set.

1 Introduction

Intrusion detection is the detection of any disallowed activities in a networked computer system. Anomaly detection is one of the most popular intrusion detection paradigms and this involves discriminating between normal and anomalous data, based on the knowledge of the normal data. Compared to traditional signature-based detection, anomaly detection has a distinct advantage over signature-based approaches as they are capable of detecting novel intrusions. However, such systems can be prone to the generation of false alarms. The Dendritic Cell Algorithm (DCA) is an Artificial Immune Systems (AIS) algorithm that is developed for the purpose of anomaly detection. Current research with this algorithm [6,4] have suggested that the DCA shows not only excellent performance on detection rate, but also promise in assisting in reducing the number of false positive errors shown with similar systems.

To date, the data used for testing the DCA have been generated by the authors of the algorithm. While this approach provided the flexibility to explore the functionality of the algorithm, it has left the authors open to the criticism that the performance of the DCA has not been assessed when applied to a more standard data set. In addition to examining the performance of the DCA, such application allows for comparison with more established techniques. For this purpose, the

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 142–153, 2008.

KDD Cup 1999 (KDD 99) data set [7] is chosen as the benchmark for evaluation, as it is one of the most widely used and understood intrusion detection data sets. This data set was originally used in the International Knowledge Discovery and Data Mining Tools Competition. During the competition, competitors applied various machine learning algorithms, such as decision tree algorithms [12], neural network algorithms [10] and clustering and support vector machine approaches [2]. In addition to these traditional machine learning algorithms, a range of AIS algorithms have been applied to this data set, such as real-valued Negative Selection Algorithm (NSA) [3].

The aim of this paper is to assess two hypotheses: **Hypothesis 1**, the DCA can be successfully applied to the KDD 99 data set; **Hypothesis 2**, changing the 'antigen multiplier' and the size of 'moving time windows' have the same effect on the DCA. We also include a preliminary comparison between the DCA, the real-valued NSA using constant-sized detectors (C-detector) and the C4.5 decision tree algorithm to provide a basic benchmark. This paper is organized as follows: Section 2 provides the description of the algorithm and its implementation; the data set and its normalization are described in Section 3; the experimental setup is given in Section 4; the result analysis is reported in Section 5; and finally the conclusions are drawn in Section 6.

2 The Dendritic Cell Algorithm

2.1 The Algorithm

The DCA is based on the function of *dendritic cells* (DCs) of the human immune system, using the interdisciplinary approach described by Aickelin *et al.* [1], with information on biological DCs described by Greensmith *et al.* [5]. The DCA has the ability to combine multiple signals to assess current context of the environment, as well as asynchronously sample another data stream (antigen). The correlation between context and antigen is used as the basis of anomaly detection in this algorithm. Numerous signal sources are involved as the input signals of the system, generally pre-categorized as 'PAMP', 'danger' and 'safe'. The semantics of these signals are shown as following:

- **PAMP:** indicates the presence of definite anomaly.
- **Danger Signal (DS):** may or may not indicate the presence of anomaly, but the probability of being anomalous is increasing as the value increases.
- **Safe Signal (SS):** indicates the presence of absolute normal.

The DCA processes the input signals associated with the pre-defined weights to produce three output signals. The three output signals are costimulation signal (Csm), semi-mature signal (Semi) and mature signal (Mat). The pre-defined weights used in this paper are those suggested in [5], as shown in Table 1. The equation for the calculation of output signals is displayed in Equation 1,

$$O_j = \sum_{i=0}^{2} (W_{ij} \times S_i) \quad \forall j \tag{1}$$

Table 1. Suggested weights for Equation 1

	PAMP S_0	**Danger Signal** S_1	**Safe Signal** S_2
Csm O_0	2	1	3
Semi O_1	0	0	3
Mat O_2	2	1	-3

where O_j are the output signals, S_i is the input signals and W_{ij} is the transforming weight from S_i to O_j.

The DCA introduces individually assigned migration thresholds to determine the lifespan of a DC. This may make the algorithm sufficiently robust and flexible to detect the antigens found during certain time periods. For example, in real-time intrusion detection there are always certain intervals between the time when attacks are launched and the time when the system behaves abnormally. The use of variable migration thresholds generates DCs whom sample different time windows, which may cover the intrusion intervals.

An individual DC sums the output signals over time, resulting in cumulative Csm, cumulative Semi and cumulative Mat. This process keeps going until the cell reaches the completion of its lifespan, that is, the cumulative Csm exceeds the migration threshold, the DC ceases to sample signals and antigens. At this point, the other two cumulative signals are assessed. If the cumulative Semi is greater than the cumulative Mat value, the cell differentiates towards semi-mature state and is assigned a 'context value' of 0, and vice versa - greater cumulative Mat results in the differentiation towards mature state and a context value of 1. To assess the potential anomalous nature of an antigen, a coefficient is derived from the aggregate values across the population, termed the 'MCAV' of that antigen. This is the proportion of mature context presentations (context value of 1) of that particular antigen, relative to the total amount of antigens presented. This results in a value between 0 and 1 to which a threshold of anomaly, termed 'MCAV threshold', may be applied. The chosen value for this threshold reflects the distribution of normal and anomalous items presented within the original data set. Once this value has been applied, antigens with a MCAV which exceeds this threshold are classified as anomalous and vice versa. To clarify the algorithm a pictorial representation is present in Figure 1.

2.2 The Implementation

The general function of the system is to read data instances of the data set and then output the MCAV of each type of antigens. In order to implement this function, three major components are implemented:

– *Tissue*: processes the data source to generate antigens and signals, in each iteration *Tissue* stores the antigens into random indexes of an antigen vector and updates current signals to a signal vector.

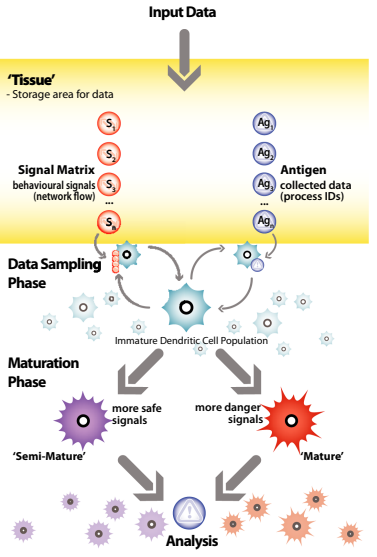


Fig. 1. The illustration of the DCA processes

- *DCell*: manages the DC population and interacts with *Tissue* to process the antigens and signals.
- *TCell*: interacts with *DCell* to produce the final results.

Two additional functions, antigen multiplier and moving time windows, are added into the system for the purpose of optimization. The DCA requires multiple instances of identical antigens, termed the 'antigen type', so processing across a population can be performed in order to generate the MCAV for each antigen type. The antigen multiplier is implemented to overcome the problem of 'antigen deficiency', that is, insufficient antigens are supplied to the DC population. As one antigen can be generated from each data instance within a data set such as KDD 99, the antigen multiplier can make several copies of each individual antigen which can be fed to multiple DCs.

The inspiration of applying moving time windows is from processes seen in the human immune system. The signals in the immune system persist over time, thus they can influence the environment for a period of time. The persistence of the signals can be presented by the cascade of signals within their affective time period. Due to missing time stamps in the KDD 99 data set, tailored window sizes for each data instance are not applicable, and a fixed window size is applied. The new signals of each iteration are calculated through Equation 2,

$$NS_{ij} = \frac{1}{w} \sum_{n=i}^{i+w} OS_{nj} \quad \forall j \tag{2}$$

where NS_{ij} is the new signal value of instance i in category j, w is the window size, and OS_{nj} is the original signal value of instance n in category j.

```
input  : antigens and pre-categorized signals
output: antigen types plus MCAV
initialize DC population;
while incoming data available do
    update tissue antigen vector and signal vector;
    randomly select DCs from DC population;
    for each selected DC do
        assign a migration threshold;
        while cumulative Csm<=migration threshold do
            get and store antigens;
            get signals;
            calculate interim output signals;
            update cumulative output signals;
        end
        if cumulative Semi<=cumulative Mat then
            cell context=1;
        else
            cell context=0;
        end
        log antigens plus cell context;
        terminate this DC and add a naive DC to the population
    end
end
while TCell analysis is not completed do
    for each antigen type do
        calculate MCAV;
    end
    log antigen types with corresponding MCAV;
end
```

Algorithm 1. Pseudocode of the implemented DCA

In brief the DCA combines multiple sources of input data in the form of pre-categorized signals and antigens. This input is processed across a population of DCs to produce the MCAV which is used to assess if an antigen type is normal or anomalous. Antigen multiplier and moving time windows are added to the algorithm to adapt the KDD 99 data set for use with this algorithm, as well as to assess the hypothesis of they having the same effect on the DCA. The pseudocode of the implemented DCA is shown in Algorithm 1.

3 The KDD 99 Data Set and Normalization Processes

3.1 The Data Set

The KDD 99 data set is derived from the DAPRA 98 Lincoln Lab data set [8] for the purpose of applying data mining techniques to the area of intrusion detection. The DAPRA 98 data set contains two data sources, which are the network sniffer data from the sniffer placed between a router and the outside gateway and the Solaris system audit data from the Solaris audit host. The KDD 99 summarizes

the two data sources into connections (data instances), each connection has 41 features (attributes), which can be grouped into four categories [11]:

- **Basic Features:** derived from the packet headers without inspecting the payload.
- **Content Features:** from the assessment of TCP packets by using domain knowledge of intrusion detection.
- **Time-based Traffic Features:** from the statistical analysis to captures the properties with a time window of two seconds.
- **Host-based Traffic Feature:** from the statistical analysis of the properties over the past 100 connections.

The KDD 99 is one of the few labeled data sets available in the field of intrusion detection. The data instances are labeled as normal connections or attack types, and the attacks can be grouped into four categories: Denial of Service (DOS), Remote to Local (R2L), User to Root (U2R) and Probe. The data set used in this paper is the 10% subset of the KDD 99 data set that is commonly used by other researchers. It consists of 494021 data instances, which are relatively massive. The whole data set would be more computational extensive, and hence much more difficult to handle, especially for the real-valued NSA with C-detector and the C4.5 decision tree algorithm. Both algorithms require training stage, the large the data set is, the longer the training would take. The 10% subset is statistically compared with the whole data set, and it features the similar ratio of the normal connections and the attacks.

3.2 Normalization of the Data Set

As anomaly detection is a two-class classification, the labels of each data instance in the original data set are replaced by either 'normal' for normal connections or 'anomalous' for attacks. Due to the abundance of the attributes, it is necessary to reduce the dimensionality of the data set, to discard the irrelevant attributes. Therefore, information gains of each attribute are calculated and the attributes with low information gains are removed from the data set. The information gain of an attribute indicates the statistical relevance of this attribute regarding the classification [11]. The information gain, termed $Gain(S, A)$ of an attribute A relative to a collection of examples S, is defined as Equation 3 [13],

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} (\frac{|S_v|}{|S|} Entropy(S_v)) \tag{3}$$

where $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v. The entropy of S relative the 2-wise classification, termed $Entropy(S)$, is defined as Equation 4 [13],

$$Entropy(S) \equiv \sum_{i=1}^{2} -p_i log_2 p_i \tag{4}$$

where p_i is the proportion of S belonging to class i.

The histograms of the remainder attributes are assessed for the normalization of the DCA, to abstract the knowledge of both normal and anomalous. Based on the characteristics of the input signals, ten numeric attributes are grouped into the categories as follows:

- **PAMP:** serror rate, srv serror rate, same srv rate, dst host serror and dst host rerror rate.
- **DS:** count and srv count.
- **SS:** logged in, srv different host rate and dst host count.

Let x be the value of an attribute, if it is certain that anomalies appear when $x \in [m, n]$, this attribute can either be PAMP or DS; otherwise if normality arises in this range, this attribute can be SS. The value of this attribute is then normalized into the range from 0 to 100 through *linear normalization* defined by Equation 5,

$$f(x) = \begin{cases} 0 & x \in [0, m) \\ \frac{x}{n-m} \times 100 & x \in [m, n] \\ 100 & x \in (n, +\infty) \end{cases} \tag{5}$$

where $f(x)$ is the normalization function. The average of the multiple attribute values in each signal category is the value of that category. In addition, the other data steam of the DCA, the antigens, are created by combining three nominal attributes, which are protocol, service and flag. Multiple instances of each antigen type can generated through this way, which satisfies the requirement of the DCA for multiple observations of each antigen type. It makes sense in the case of both human immune system and intrusion detection: since antigens with the same pathogenic patterns can invade the human immune system over and over again; and attacks with the same patterns can be launched discretely over time in a networked computer system.

The ten attributes selected for the signals in the DCA are chosen to represent the detectors and antigens in the NSA. These attributes are normalized into the range from 0 to 1, using *max-min normalization*, thus the data space is a unitary hypercube $[0, 1]^{10}$. The data set is then rearranged to generate ten subsets through 10-folder cross-validation. The training data is made of the nine folders and the testing data is made of the one folder in each subset. The self set of the NSA is derived from all the normal data instances in the training data, and the antigens are the data instances in the testing data. The input data of the C4.5 decision tree algorithm contains the same attributes as those of the NSA but without normalization, and the labels of normal and anomalous are provided for the purpose of training.

4 Experimental Setup

Both the DCA and the NSA are implemented in C++ with the g++ 4.2 complier, and the C4.5 decision tree algorithm is performed in Weka [14], which is a collection of machine learning algorithms for data mining tasks. The experiments

are run on a PC on which Ubuntu Linux 7.10 with a kernel version of 2.6.22-14-generic is installed. The receiver operating characteristics (ROC) analysis is performed to evaluate the classification performance of the DCA and the NSA. The true positive (TP) rate, false positive (FP) rate, true negative (TN) rate and false negative (FN) rate of each experiment are calculated, and the relevant ROC graphs are plotted as well. Three sets of experiments are performed: various DCA versions (E1), the real-valued NSA using C-detector (E2), and the C4.5 decision tree algorithm (E3).

In all experiments related to the DCA, the size of the DC population is set as 100 and it is constant as the system runs. The migration threshold of an individual DC is a random value between 100 and 300, to ensure this DC to survive over multiple iterations. The 'perfect MCAV' of an antigen type is calculated based on the labels of the original data set, normal is equivalent to context value 0 and anomalous is equivalent to context value 1. To generate the classification results of the DCA and the 'perfect classification results' from the perfect MCAVs, a MCAV threshold of 0.8 is applied. The MCAV threshold is derived from the proportion of anomalous data instances of the whole data set , which is equal to 80%. The classification results of the DCA are then compared with the perfect classification results, to assess the TP, FP, TN and FN. Three experiments of E1 are performed corresponding to the DCA versions as following:

- **E1.1:** the basic version of the DCA.
- **E1.2:** the system with antigen multiplier, the antigens are multiplied by 5, 10, 50 and 100.
- **E1.3:** the system with moving time windows, the window size is respectively equal to 2, 3, 5, 7, 10, 100 and 1000.

For each single experiment, ten runs are performed and the final result is the average of the ten runs. In order to make the results from different experiments more comparable, a fixed sequence of random seeds for ten runs is used. For E1.2 and E1.3, the two-sided Mann-Whitney test is performed to assess if various parameters can make the results statistically different from each other. The statistical significance α is set as 0.05, thus giving a confidence of 95% to either accept or reject the null hypothesis.

E2 includes a range of experiments of the NSA, as the data space increases from two dimensional to ten dimensional. According to the parameters mentioned in [9], the self radius is equal to 0.1 and the detector amount is increased to 1000 because of the large size of the data, and the matching rule used is the Euclidean distance matching. The results produced by the algorithm are compared to the labeled testing data, namely the 'perfect result', to perform the ROC analysis. The final results of each dimension is the average of ten subsets. The experiment setup of C4.5 decision tree algorithm are as follows: the classifier chosen in Weka is J48, which is a class for generating an unpruned or a pruned C4.5 decision tree; the test option of the classification is set as 10-folder cross-validation.

Table 2. The ROC results of the experiments in E1

Category	Parameter	TP Rate	TN Rate	FP Rate	FN Rate
E1.1	-	0.7375	1	0	0.2625
E1.2	5	0.75	1	0	0.25
E1.2	10	0.74375	1	0	0.25625
E1.2	50	0.75	1	0	0.25
E1.2	100	0.75	1	0	0.25
E1.3	2	0.75	1	0	0.25
E1.3	3	0.75	1	0	0.25
E1.3	5	0.74375	1	0	0.25625
E1.3	7	0.75	1	0	0.25
E1.3	10	0.75625	1	0	0.24375
E1.3	100	0.71875	0.96	0.04	0.28125
E1.3	1000	0.7	0.979592	0.0204082	0.3

Table 3. The ROC results of the experiments in E2

Data Dimension	TP Rate	TN Rate	FP Rate	FN Rate
2	0.98367	0.42944	0.37055	0.01633
3	0.23462	0.71834	0.08165	0.76538
4	0.08971	0.79289	0.00711	0.91029
5	0	0.79993	0.00007	1
6	0	1	0	1
7	0	1	0	1
8	0	1	0	1
9	0	1	0	1
10	0	1	0	1

5 Result Analysis

The results of E1 are shown in Table 2, which indicate the antigen multiplier cannot consequentially enhance the system performance. The signals associated with the misclassified antigens are generated incorrectly from the original data set, thus the DCs always assign wrong context values no matter whether the antigens are multiplied or not. Moreover, the moving time windows cannot significantly improve the system performance either. Due to the limitation of the data set, the tailored window sizes of each data instance that may result in better system performance are not applicable. Furthermore, the Mann-Whitney test suggests a 95% confidence to accept the null hypothesis, that is, the results of all the experiments in E1 are not statistical different from each other.

The results of E2 are shown in Table 3, and the ROC results E2 from two dimensional to ten dimensional are shown in Figure 2. The algorithm produces acceptable results when the data space is two dimensional. But as the dimensionality increases, the classification performance is getting worse and worse. The algorithm cannot detect any anomalies when the the data space is six

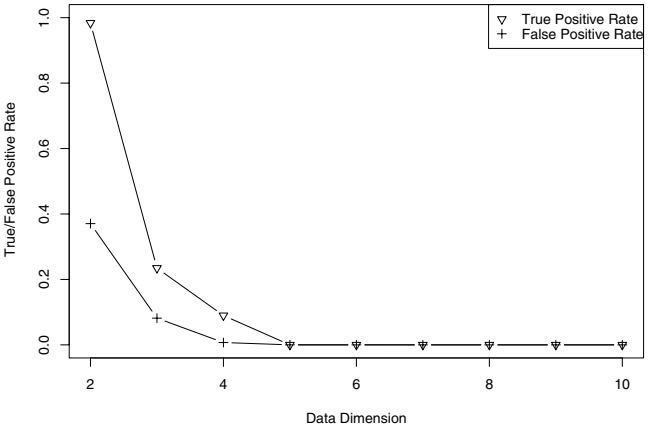


Fig. 2. Results of the real-valued NSA with C-detector across different dimensionality

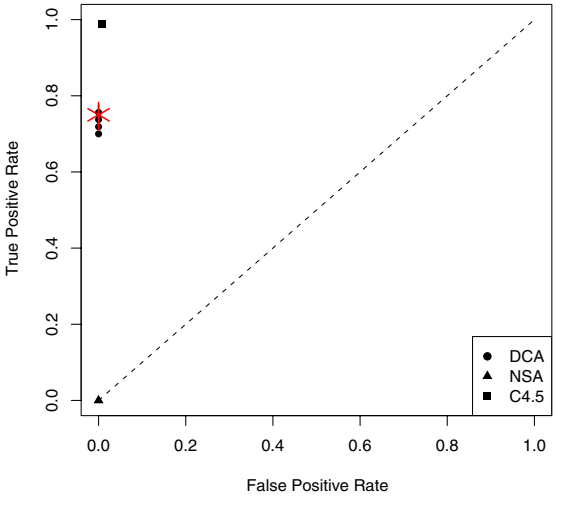


Fig. 3. The ROC graph of E1, E2 and E3 as data space is ten dimensional

dimensional or more. As the dimensionality of the data space increases, the search space grows exponentially, thus it is becoming more and more difficult to generate sufficient detectors that can effectively cover the space of non-self.

The ROC graph of the results in E1 and E2 when the dimensionality is ten is shown in Figure 3. The results of the DCA are located on the top-left corner of the graph, showing that all versions the DCA can successfully detect around 75% true anomalies over all actual anomalies as well as produce no or few false alarms. The real-valued NSA with C-detector cannot produce any useful results, as it fails to detect any anomalies. Moreover, as expected the C4.5 decision tree algorithm produces superb results, the true positive rate is 0.988 and the false

positive rate is 0.008. This algorithm is designed specifically for the purpose of data mining, its classification performance is supposed to be better than the other two algorithms that are designed for the purpose of anomaly detection. But in terms of false positive rate, the classification performances of the DCA and the C4.5 decision tree algorithm are comparable with each other.

6 Conclusions and Future Work

This paper presents the algorithm behaviors of the DCA, when it is applied to a standard data set, the KDD 99 data set. The results show that the DCA is able to work with the data set and produce reasonable performance, therefore **Hypothesis 1** is accepted. Moreover, the DCA is an unsupervised learning algorithm, it does not require training with normal data instances. It acquires the knowledge of normal and anomalous through the categorization of signals based on basic statistical analysis. Besides, it is not constrained by high dimensionality of the data sets. Thus the DCA is applicable to large data sets with high dimensionality. The real-valued NSA with C-detector has poor classification performance on the high dimensional KDD 99 data set, it could not manage to detect any anomalies as the dimensionality increases up to six or more. Therefore, this algorithm is not applicable to the data sets with high dimensionality. As a specialized machine learning algorithm, the C4.5 decision tree algorithm produces excellent results, it provides a benchmark showing the ideal results of the KDD 99 data set.

Due to limitations of the data set, the DCA could not be optimized by either antigen multiplier or moving time windows. First of all, it is only possible to generate one unique antigen from each data instance, leading to the insufficient observations of each antigen type by relative DCs, the problem cannot be solved with the antigen multiplier. Furthermore, the time stamps of each connection are unavailable, thus it is impossible to apply tailored window sizes in the system, and hence the advantage of the moving time windows is not fully utilized. Even though, both antigen multiplier and moving time windows have the same effect on the DCA for this particular data set, and hence **Hypothesis 2** is accepted.

Some future directions of DCA research can be: first of all, to perform more rigorous comparisons between the DCA and other AIS algorithms; Secondly, to apply the DCA to other data sets, to further explore the limits of the DCA and to understand the antigen multiplier and moving time windows; Thirdly, to add more features to the DCA, to make the algorithm more adaptive and flexible.

References

1. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger Theory: The Link between AIS and IDS. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
2. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: Detecing intrusions in unlabled data. In: Barbara, D., Jajodia, S. (eds.) Applications of Data Mining in Computer Security, ch. 4. Kluwer, Dordrecht (2002)

3. Gonzalez, A., Dasgupta, D.: Anomaly Detection Using Real-Valued Negative Selection. Genetic Programming and Evolvable Machines 4(4), 383–403 (2004)
4. Greensmith, J., Aickelin, U.: DCA for SYN Scan Detection. In: Genetic and Evolutionary Computation Conference (GECCO), pp. 49–56 (2007)
5. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
6. Greensmith, J., Twycross, J., Aickelin, U.: Articulation and Clarification of the Dendritic Cell Algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 404–417. Springer, Heidelberg (2006)
7. Hettich, S., Bay, S.D.: The UCI KDD Archive. Technical report, University of California, Department of Information and Computer Science, Irvine, CA (1999), http://kdd.ics.uci.edu
8. MIT Lincoln Lab Information System Technology Group. The 1998 Intrusion Detection Off-line Evaluation Plan (March 1998), http://www.ll.mit.edu/IST/ideval/data/1998/
9. Ji, Z., Dasgupta, D.: Applicability Issues of the Real-Valued Negative Selecion Algorithms. In: Genetic and Evolutionary Computation Conference (GECCO), pp. 111–118 (2006)
10. Kayacik, N., Zincir-Heywood, G., Heywood, M.: On the Capability of an SOM based Intrusion Detection System. In: Proceedings of International Joint Conference on Neural Networks, vol. 3, pp. 1808–1813 (2003)
11. Kayacik, N., Zincir-Heywood, G., Heywood, M.: Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 1999 Intrusion Detection Datasets. In: Third Annual Conference on Privacy, Security and Trust (PST) (2005)
12. Levin, I.: KDD 1999 Classifier Learning Contest: LLSoft's Results Overview. SIGKDD Explorations 1(2), 67–75 (2000)
13. Mitchell, T.M.: Machine Learning. McGraw-Hill Series in Computer Science. McGraw-Hill, New York (1997)
14. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Evaluation and Extension of the AISEC Email Classification System

Nrupal Prattipati and Emma Hart

Napier University, Scotland
e.hart@napier.ac.uk

Abstract. An existing system - AISEC - which categorises email as interesting or uninteresting using an immune-inspired algorithm is implemented as a plug-in to Outlook to allow seamless user testing. Experiments are performed with a new, large data set to validate previous published results. We show comparable results can be obtained on different data-sets *if* the system parameters are correctly tuned; the algorithm is particularly sensitive to certain parameters. Some flaws in the original algorithm are identified; a modification is proposed to the learning process of the algorithm and to the mutation operator. Tests with the modified algorithm in a number of scenarios in which users' interests frequently change show the improved algorithm is capable of continuously adapting to achieve high classification accuracy and can accurately track changes in user interests. The improvements are statistically significant when compared to the original system.

1 Introduction

According to a survey carried out by White Collar Productivity Index [9], in 2004 a person spent on average 8.8 hours a week handling email. The survey showed that even if all spam is removed, a significant amount of time is spent reading and subsequently ignoring emails that we have no interest in. For example, emails from a Faculty administrator regarding timetabling information are of no interest to a lecturer on sabbatical but are not "spam" in the classical sense. Users' interests change continuously change however; at the end of the sabbatical period, the emails may become of interest again. On the other hand, a lecturer may lose interest in emails relating to conference during semester due to a busy teaching timetable. It is important to distinguish the difference between spam and mail that is simply uninteresting; spam refers to unsolicited mail that at no time is of interest to the user. Many highly specialised pieces of software exist for identify and removing spam email — this is not the concern of the application described in this paper. Indeed, spam filtering necessitates that the false positive rate[1] of a filter must be minimised, as the consequences of mis-classifying and removing legitimate email from a user's inbox can be enormous.

In 2003, Secker *et al* published a system named AISEC which was capable of classifying emails as interesting or non-interesting and removed un-interesting

[1] Incorrectly identifying legitimate mail as spam.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 154–165, 2008.

mail from a user's inbox. Furthermore, the system was shown to be capable of continuous learning; following changes in a user's interest, the system could adapt to the new interests. Results were published from a single set of 2268 emails of which 32.7% were classified as uninteresting and the remainder interesting. The results were compared to performance on the same data set with a naïve Bayesian system; although performance was similar overall, AISEC showed improved performance during certain periods of time. It was postulated that this was due to the ability of AISEC to adapt to changes in the data, such as a word used commonly in uninteresting emails suddenly beginning to appear in interesting email. However, this hypothesis was never explicitly tested by examining the data in detail or by testing specific scenarios in which emails were known to change in content. Therefore, in this paper we specifically address the following objectives:

- to validate published results by running system on a completely different set of test emails - this is partly due to the unavailability of original set, but furthermore, offers an opportunity to investigate the sensitivity of the algorithms parameters to different data sets.
- to explicitly test the ability of the algorithm to adapt to changing interests, by setting up a number of test scenarios in which the users interest in emails from a particular source changes from interesting to un-interesting (and vice-versa) over a period of time.
- to provide a number of modifications to the algorithm which improve the speed at which it adapts and the overall accuracy of the classification algorithm.

The remainder of the paper is organised as follows. Section 2 gives an overview of the existing algorithm. Section 3 describes the re-implementation of AISEC as an Outlook plug-in and the experimental test-bed used to investigate its performance. Section 4 reports results on experiments designed to validate the performance of the existing algorithm. This is followed by a description of two extensions to the existing algorithms in sections 5 and 6, including presentation of new results. We conclude with some remarks on the use of immune algorithms in a continual learning scenario and recommendations for future work.

2 AISEC

The algorithm is presented briefly — this description is taken directly from the original publication of Secker *et al.* For pseudo-code and implementation details of the algorithm, the reader is referred to [8].

Essentially, *AISEC* classifies mail into two classes, semantically labelled as 'interesting' or 'uninteresting'. The algorithm is inspired by some of the properties observed in the natural immune system. It uses inspiration from B-cells to represent one class of data, that of uninteresting emails. A B-cell consists of a set of words derived from the subject and sender fields of a training set of uninteresting emails and occurs in one of two states - those that are naïve potentially

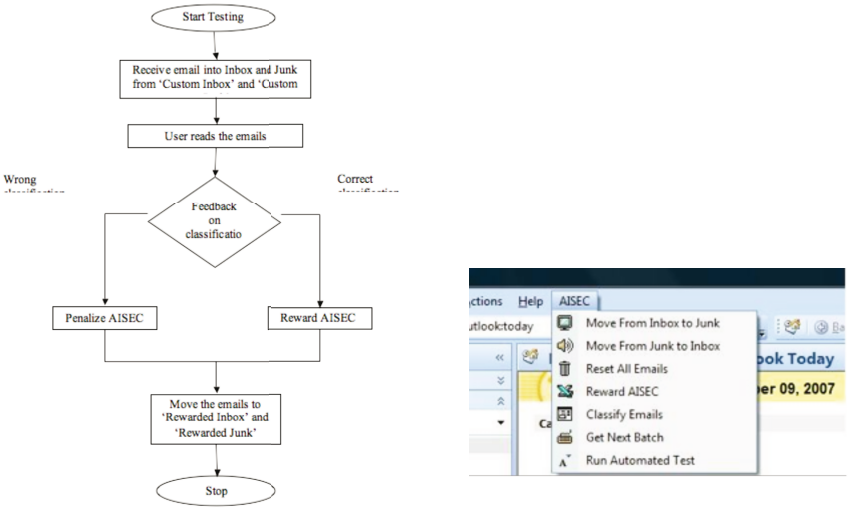
classify uninteresting mail, those that are mature have received co-stimulation from a user confirming their classification and are regarded as memory cells. Gene libraries are used to store words that have been used in uninteresting mail and are used to mutate existing B-cells. Cells become stimulated via matching process which matches words in the B-cell against incoming emails. B-cells become stimulated once their stimulation level exceeds a pre-defined threshold. Cloning of high-affinity existing cells is used to generate new cells - the original authors note the impractically of generating random detector cells given the size of the search space, therefore all new naiv̈e cells are mutants of existing cells. The algorithm is dynamic - user feedback and cloning continuously produce new cells. In order to prevent unlimited population growth, a cell-death process is implemented in which cells which do not receive sufficient stimulation over a period of time are purged from the system. User feedback is given asynchronously to aid classification but on a regular basis. As the algorithm is designed to address concept drift over long periods, the design of the algorithm ensures that reasonable pauses in this feedback do not cause an undue drop in classification accuracy.

A number of other immune-inspired approaches to email classification exist in the literature. In the main, these relate to the classification of email into spam and non-spam sets. As noted in the introduction, this task has signifcantly different requirements from classifying mail as interesting/not-interesting — for example, while a user's interest in a particular topic may wax and wane over time, their interpretation of what is spam email is likely to remain constant.The reader is referred to the works of Yue [10] and Oda [6] for examples of immune-inspired spam classification systems and also to [5] for related work in the domain of adaptive information filtering (using documents rather than emails).

3 Experimental Approach

The AISEC system was converted to .NET using Visual Studio Tools for Office. New buttons and new folders are added to Microsoft Outlook 2007 using an add-in project. The add-in allows seamless integration of AISEC with Outlook such that the system can run in real-time. The new system is referred to as AISEC-Outlook to distinguish it from the original.

The working of the Microsoft Outlook add-in is shown in 1(a). Whenever a new email is downloaded from the email server, AISEC-Outlook classifies the email as interesting or uninteresting for the user and places in an appropriate folder. As the user reads the email feedback is taken from the user depending on his/her actions. There are two kinds of feedback that can be given to AISEC-Outlook, positive, for correct classification of the email and negative, implying incorrect classification of the email. In the case of positive feedback, the user is not required to do anything except read the email, upon which Microsoft Outlook 2007 marks the email as read by assigning the unread property of the email item to false. AISEC-Outlook identifies such emails and assumes positive feedback. When AISEC-Outlook has mis-classified an email, the user is required



(a) Overview (b) Additions to Outlook

Fig. 1. An overview of AISEC used as an add-in Microsoft Outlook, showing the testing procedure and illustrating the integration of the system into Outlook for testing

to move the email to its correct folder. AISEC-Outlook recognizes the moved emails and interprets this as negative feedback. Based on this feedback, AISEC-Outlook rewards the BCells. In the case of positive feedback, all the BCells with a higher affinity than the affinity threshold for the correctly classified email are stimulated and the one with the highest affinity is cloned and mutated to generate new BCells belonging to the next generation. In the case of negative feedback, all the BCells with an affinity above the affinity threshold for the mis-classified email are removed, and the words from subject, sender and return address fields are removed from the repository of words for each field. This avoids further mis-classification of similar emails. If an email which is uninteresting is mis-classified as interesting then, the emails words in subject, sender and return address fields are added to the repository of the words. The email is added to nave BCells. This avoids further mis-classification of such kind (similar in topic) of emails. This process is repeated for every incoming email and the system adapts to the users interests if there has been a change.

Providing feedback for every email in a large test set during testing would be time consuming and a tedious process. Therefore all the stages in classifying an email are automated. New buttons and folders are created in Outlook. Identifying a single email which has been read by the user in a folder requires iterating through the entire set of emails, which is time and processor consuming process. To avoid this, new folders are added and at each stage in the process of classifying the email the email is moved between the folders. At any time a particular

folder only contains the emails that need to be classified or taken feedback from. Figure 1(b) illustrates the buttons that have been added to facilitate testing and classification of emails.

3.1 Methodology

A test set of 2412 emails are used gathered during a period of approximately 9 months. The email set consisted of personal email plus emails that were received after deliberately subscribing to a number IT related sites such as WebProNews [1], and a set of sites advertising job vacancies, e.g. IT Job Board, Jobsite Vacancy etc. Of these emails, 70.81% are classified by hand as interesting and 29.19 % were classified as uninteresting. As in the original paper, the temporal ordering of emails within the test was preserved and only the words contained in the subject and sender fields of the e-mail were used. The sender information also included the return address, as these fields may differ. The fields were tokenized using spaces and the characters ., ,, (,), !, @, ¡, ¿ as delimiters and each token inserted into a separate element of the correct feature vector. Simulated user feedback was given to both algorithms after the classification of each e-mail.

The standard measure of classification accuracy was used to evaluate results, i.e, the percentage of all emails correctly classified. However, as the data set is unbalanced in terms of the number of samples in each class, a confusion matrix [7] was also used to examine the recognition rate of each individual class. We additionally report the *precision* and *recall* for each test using standard definitions of these terms.

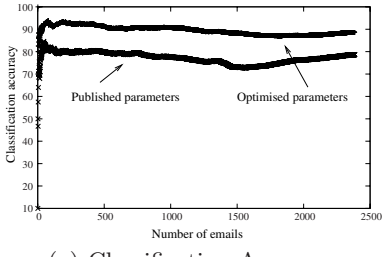
Table 1. Comparison of predictive accuracy of AISEC-Outlook with original parameters and optimised parameters (standard deviation given in brackets)

	Accuracy	Precision	Recall
Original Test Set	89.09 (0.97)	82.20	81.13
New Test Set (original parameters)	78.68 (2.00)	99.67 (0.04)	70.43 (2.80)
New Test (optimised parameters)	88.61 (1.27)	99.58 (0.07)	84.44 (1.81)

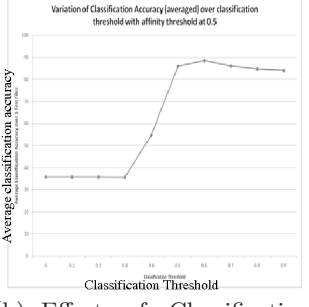
4 Results: Validation

The system was trained with a training set of 25 emails — the first 25 uninteresting ones. The system was first run with the parameters specified by Secker in [8]; as the system is stochastic, 10 runs were performed in order to derive an average classification accuracy. This however produced disappointing results, reported in lines 1 and 2 of table 1. A mean classification accuracy of $78.68\% \pm 0.04$ was obtained, compared to that of $89.09\% \pm 0.97$ reported by Secker. Interestingly however, the *precision* obtained on the new dataset is significantly higher than originally reported; this occurs however, at the expense of *recall* in the system, which shows a considerable drop over the figure reported in [8]. Following this,

extensive experimentation was performed in which the settings of two parameters were investigated. The *affinity threshold*, which determines the minimum affinity between two compared cells required to cause stimulation of a cell, and the *classification* threshold, which determines whether or cell is ultimately classified as interesting or uninteresting. Lack of space prevents all results being presented here; however, test results showed that the mean accuracy could be significantly improved by tuning the parameters. Figure 2(b) shows the results of one experiment in which the effect of the classification threshold parameter was investigated; clearly the system is particularly sensitive to this value. The final classification accuracy obtained with optimised parameters is shown in table 1 - this was obtained with the classification threshold set to 0.3 and the affinity threshold 0.5. Once again, the *precision* of the optimised system is very high; the optimised parameters lead to an increase in *recall* in the system, and overall to a classification accuracy similar to that reported in [8].



(a) Classification Accuracy



(b) Effect of Classification threshold

Fig. 2. Tests showed that performance of AISEC on the new data set could be increased by optimising the parameters. In particular, the classification threshold parameter significantly affected the results.

Figure 2(a) shows the trends in accuracy for the original reported parameter set as the optimised set as emails are presented to the system. At each datapoint, the classification accuracy was measured following presentation of each email, taking into account all previous classifications. The graph shows that although both systems stabilise quickly to a high classification accuracy, the optimised parameters perform significantly better (this has been verified by a t-test). However, with a dataset which is biased towards one class, the classification accuracy does not necessarily reveal the whole picture. Therefore, confusion matrices are presented in figure 3(a) and 3(b). The matrices clearly show that AISEC produces a very low false positive rate[2] — 0.59% in the original parameters and 1.03% in the optimised set. The false negative rate however is higher in both cases (26.32% and 13.25% respectively), suggesting that the system has a

[2] Uninteresting mail classified as interesting.

tendency to wrongly classify interesting mail as uninteresting. In an email classification application, this could lead to interesting mail being ignored by the user.

Overall, the results serve a useful validation of AISEC published in [8]. Although we are unable to obtain the details of the email set used in [8], it seems reasonable to assume that the content of the email set used in our experiment is largely different. Nevertheless, the algorithm performed with comparable accuracy, and in fact, provided higher precision than originally described. However, our results emphasises the perhaps obvious necessity to tune the system to a particular users inbox. Results are likely to vary widely depending on the precise nature of the relationship between words in a particular inbox and in the frequency and nature of the changes in interest. We return to this later in the paper.

	Predicted	
Actual	uninteresting	interesting
uninteresting	675	4
interesting	226	1480

(a) original parameters

	Predicted	
Actual	Uninteresting	interesting
uninteresting	672	7
interesting	226	1480

(b) optimised parameters

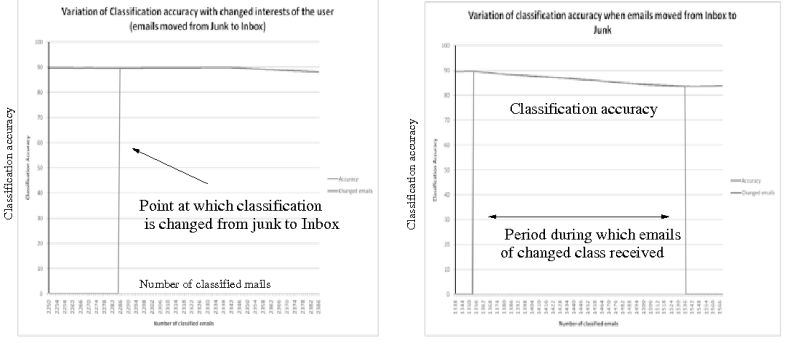
Fig. 3. Confusion matrices obtained from a single run of the system with the originally specified parameters and optimised parameters

5 Extension 1: Adaptability of AISEC: Reacting to Changing Interests

Although results in [8] show that AISEC capable of continuous learning, and potentially of tracking concept drift, no results have been published in which *explicit* changes in user interest are tested. Therefore, we designed tests to verify whether explicit changes in a user's interests could be tracked. Two tests are performed; in the first, emails from the source WebProNews [1], currently designated as uninteresting, are designated by the user to to be interesting and therefore moved to the user's Inbox from the Junk folder. In the second test, the reverse operation is performed; the user's interest change once more and new emails in the Inbox from WebProNews are moved back to the Junk folder.

Figures 4(a) and 4(b) show the effect on the classification accuracy over time as emails are classified. Figure 4(a) clearly shows that AISEC adapts quickly to a change in interest from Junk to Inbox. Figure 5 shows the testing procedure used by the AISEC algorithm: when AISEC classifies the first email from the changed as uninteresting, the user supplies negative feedback. Based on this feedback, AISE penalizes all the BCells responsible for the recognition of the email and removes the BCell with the highest affinity. This has the overall effect

of reducing the affinity of the BCells with the type of emails in which the user is now interested. Upon repetition of this process, all the BCells with high affinity towards the interesting emails are eventually removed. As reflected in the graph, AISEC adapts to the interests of the user. However, figure 4(b) shows an obvious decrease in classification accuracy when the reverse process occurs: when the user's *loses* interest in a topic, AISEC fails to react. Further examination of the algorithm presented in figure 5 reveals the reason: the verification of the classification by the user only happens when an email has been classified as uninteresting. Therefore, mails wrongly classified as interesting by AISEC do not receive user feedback. AISEC was modified accordingly: when the user supplies negative feedback from mis-classification of an item in the Inbox, the email is now added to the repository of B-Cells responsible for classifying mails. This new B-Cell will recognise any further remains of this type, increasing its stimulation and causing the cell to be cloned.



(a) Junk to Inbox (b) Inbox to Junk

Fig. 4. Ability of AISEC to adapt to changes: the figures show change in classification accuracy in two scenarios. In the left-hand figure, the user gains interest in a set of emails - the top most line indicates classification accuracy. In the right-hand figure, the user now designates a set of emails as uninteresting.

Results of Modified Algorithm. To test the modified algorithm, a further experiment was run: 265 emails received from the source WebProNews were moved from the Inbox to Junk after 1200 emails has been classified. At iteration 1600 (i.e after 1600 email classifications), 143 emails that had been received from WebProNews during this interim period were moved back to the Inbox. Figure 6 shows the classification accuracy against number of emails received obtained using the modified algorithm; the point at which the emails which undergo a change in classification are highlighted on the graphs: the figure shows that the modified version of AISEC is now able to adapt to all changes in the user preferences, i.e. in both directions.

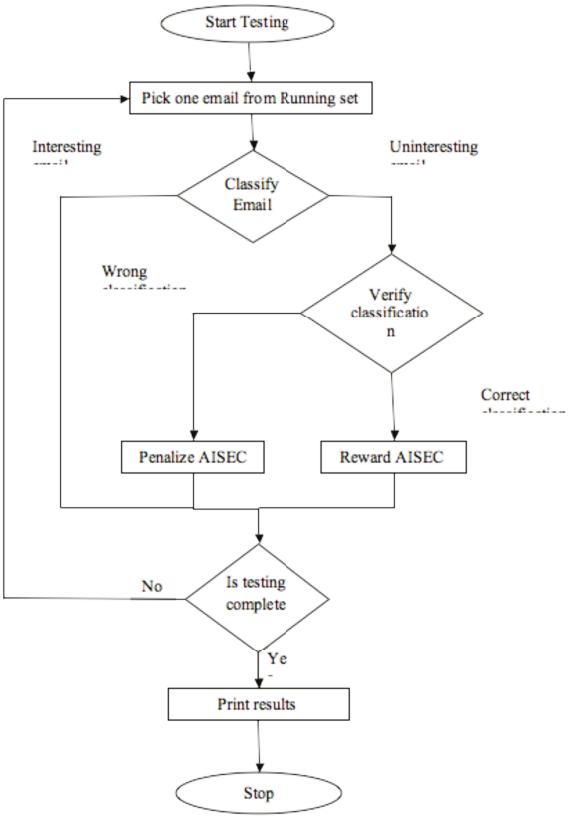


Fig. 5. The classification and reward procedure used in the original published version of AISEC. A flaw exists in that mails classified incorrectly as interesting are never penalised.

6 Extension 2: Mutation

The original version of AISEC employs a simple mutation operator for mutating B-cell vectors, which selects a random word in the feature vector and replaces it with a random word from the gene library store. However, a vast array of literature is available in the document/text analysis domain, which provides inspiration for refining this operator to better exploit relationships and semantics between words in a vector. One simple such suggestion is described here.

Word collocation (e.g. [3]) is a technique used in computational linguistics and text analysis which utilises the facts that words that are related to each other are located within a fixed distance of one another in a text. Thus, in a text about immunology, infection and pathogen are likely to occur in (say) a distance 10 from each other. B-Cells are created in the first instance by scanning an email's subject and sender lines, placing words from these fields into

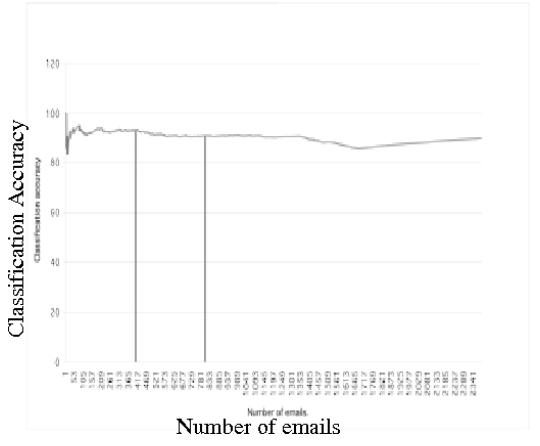


Fig. 6. The modified system is capable of maintaining classification accuracy as the user preferences for a particular set of emails change from interesting to uninteresting (first vertical line) and then back to interesting again some time later (2nd vertical line)

a vector. The subsequent ordering of words in the vector reflects the order they were originally encountered in the email; word collocation suggests that words located next to each other in the resultant vector may be related in subject. For example, a university administrator sends a mail"timetable for software engineering course" which results (after removal of stop-words) in a vector [timetable software engineering course] in which the words *software* and *engineering* are located consecutively. The original mutation operator described in [8] randomly selects a single word from the vector and replaces it with a random word from a gene library. However, we propose that the positional bias between words in the B-cell vector can be exploited during mutation by use of a positionally biased mutation operator. Therefore, we replace the mutation vector currently used in AISEC with a position-biased mutation operator proposed by Kelsey and Timmis in [4] and later analysed theoretically in [2]. This operator, known as "hot-spot" mutation, selects a position in the vector at random, and then applies mutation to a contiguous region of the vector starting from this position, thereby increasing the chance of replacing collocated words. The size of the region is fixed at 2, and the operator selection a starting position between position 0 and (n-1) in the vector, with no wrap-around. Although simplistic, we postulate that this may be particularly useful in speeding up adaptation of the B-cell repertoire after changes in a user's interests. For example, following on from the example given above, the academic in question may stop teaching a particular module and hence no longer be interested in emails concerning the software engineering degree. The proposed mutation operator increases the chances of the words *software* and *engineering* being replaced in a single application of the mutation operator.

The operator is updated as described. The modified version of AISEC was again re-run 10 times on the test set, and the classification accuracy recorded as each new email was classified. The results are shown in figure 7 which compares the performance of the original algorithm and the hot-spot modified version. Although the modified operator appears to be less effective when the algorithm has only been exposed to a small number of email, its effectiveness increases and it is seen to maintain a more consistent classification accuracy than the original operator. T-test confirmed that the final classification accuracy after presentation of the emails was significantly better with the modified algorithm.

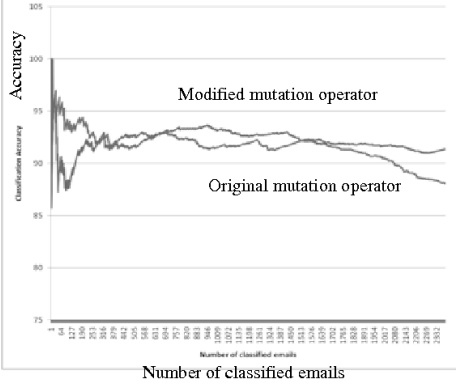


Fig. 7. Comparison of modified mutation operator to original operator

7 Conclusion

The paper has presented further experimental evidence that an immune-inspired learning algorithm, AISEC, is capable of performing continuous learning in a dynamic environment. Our research firstly provides verification of previously published results by testing the performance of the algorithm on a new test set of data; for any algorithm to become acceptable it is crucial that it is benchmarked on a number of problem data-sets. The context of in which AISEC applied — a continuous learning environment — renders existing publically available email datasets unsuitable as traditionally these are designed for use with one-shot learning algorithms. As personal mailboxes are likely to differ greatly from one person to another it was reasonable to hypothesise that the parameters of AISEC would need to be tuned in order to work with a new dataset. Although this did prove to be the case, AISEC proves to be a relatively robust algorithm, with many of the parameters robust to changes in them. The key parameter from the point of view of tuning appears to be the classification threshold. Secondly, we have explicitly tested the ability of AISEC to adapt to changing interests for the first time. Results obtained in these scenarios necessitated a modification to the algorithm to cope with changes from interesting to uninteresting. Finally,

we have further improved the algorithm to exploit the positional bias present in the B-cell representation which produced additional increases in classification accuracy.

Whilst the system is relatively simplistic as a machine learning tool and lacks the theoretical basis that perhaps make the use of more traditional learning techniques more seductive, the results suggest that the system is worthy of further exploration. An obvious avenue for future extensions is to investigate using the body of the mail in the classification - this is likely to require more sophisticated processing of the body text before including in a B-cell in order to maintain tractability. More generally, the results suggest that immune-inspired learning algorithms may have a role to play in tackling continuous learning problems.

References

1. www.webpronews.com
2. Clarke, E., Hone, A., Timmis, J.: A markov chain model of the b-cell algorithm. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 318–330. Springer, Heidelberg (2005)
3. Gledhill, C.: Collocations in Science Writing. Gunter Narr Verlag Tübingen, Tübingen (2000)
4. Kelsey, J., Timmis, J.: Immune inspired somatic contiguous hypermutation for function optimisation. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724. Springer, Heidelberg (2003)
5. Nanas, N., De Roeck, A., Uren, V.: Immune inspired adaptive information filtering. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 418–431. Springer, Heidelberg (2006)
6. Oda, T., White, T.: Immunity from spam: an analysis of an artificial immune system for junk email detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 2289–2776. Springer, Heidelberg (2005)
7. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing classifiers. In: Proceedings of the Fifteenth International Conference on Machine Learning, ICML 1998 (1998)
8. Secker, A., Freitas, A.A., Timmis, J.: Aisec: an artificial immune system for email classification. In: IEEE Congress on Evolutionary Computation CEC 2003, pp. 131–138 (2003)
9. Sherman, B.: Where do white collar workers actually spend their time in the office (2007), http://www.expertclick.com/NewsReleaseWire/default.cfm?Action=ReleaseDetail&ID=11164
10. Yue, X., Abraham, A., Chi, Z.-X., Hao, Y.-Y., Mo, H.: Artificial immune system inspired behaviour-based anti-spam filter. Soft Computing - A Fusion of Foundations, Methodologies and Applications 11(8), 729–740 (2007)

Dynamic Polymorphic Agents Scheduling and Execution Using Artificial Immune Systems

Leonardo M. Honório, Michael Vidigal, and Luiz E. Souza

Federal University of Itajubá, Brazil
{demello,michael,edival}@unifei.edu.br

Abstract. When a set of heterogeneous agents is considered to solve different kinds of problems, it is very challenging to specify the necessary number of elements, which functionally of each one will be used and the schedule of these actions in order to solve these problems. To deal with scenarios like this, the present article suggests an innovation at the Intelligent Agent Theory, a new concept called Dynamic Polymorphic Agent (DPA). This approach implies on the dynamic generation of one agent, built from the cooperation of existing agents and specific to fulfill the demanding task. To create this new entity, a monitor identifies and reads information regarding the functionalities of available agents present in the scene and, when a new problem is presented, it generates a task list to solve it. This list and the agents whose functionalities are necessary to solve the problem generate the new polymorphic agent. To fulfill this approach, two major paradigms are used: Aspect-Oriented Program (AOP) and Artificial Immune System (AIS).

Keywords: Polymorphic Agent, IA Planning, Artificial Immune Systems (AIS), multi-agent systems, Aspect-Oriented Program (AOP).

1 Introduction

An agent is, in a simple way, a premolded component that, given an input, executes a processing and generates a result. From this general concept, the agents' theory is applied to a large spectrum of problems: systems optimization, robotics, and business procedures [1]-[2], among others. This success is greatly due to the capacity to efficiently define the input information and the result generated by the processing of the agent. This predefinition of variables and functionalities generates, among other benefits, great portability, reliability and modularity to this paradigm. However, the agents, even those with the capacity of knowledge analysis and generation, are fixed functionality structures. Thus, even if the agents are capable of altering their capacity to iterate mutually or with the environment, what each element does is necessarily what it has been projected to do. In order to expand this paradigm, this work suggests a new methodology named Dynamic Polymorphic Agent (DPA).

The idea of polymorphism is not a new one. Inside the theory of analysis and object-oriented programming, which appeared in the 80's, polymorphism is a concept used to avoid implementation redundancy and to increase the reutilization of nuclear elements. In this concept, basic components are implemented and used by more

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 166–175, 2008.

complex elements through inheritance. The new derived elements inherit the properties and methods of many basic classes, thus making it possible to hold new functionalities. Even though it is flexible, polymorphism is a static concept, that is, it is only implemented during the process of problem modeling, and, once it is operating, the derived classes stay static in relation to new properties and methods.

A more flexible concept, Aspect Oriented Programming [3]-[4], appeared in the late 90's as a necessity to solve problems related to the entanglement of methods and functionalities in highly complex systems. The presence of functionalities that cut across many classes leaves the system with low reutilization and difficult modularization. Thus, the idea of aspect is one more level of abstraction, that is, the visualization of parts of the problem as being general, almost to the algorithm level, and that may be attributed to a given agent that has a specific signature during run time. Therefore this paradigm changes the systems' structure and associations during run-time, enabling to use new functionalities added into the scenario without any adjust.

However, to use this paradigm in multi-agents situations, the agents must be built over a framework that enables the access to their *Hyperdata*. The *Hyperdata* is defined here as a data set that combines the metadata – which is responsible to inform a list of all methods with their return data type, input parameters, etc – and PDDL [5] scripts that give a full description of how each method must be used, including their pre-conditions, needed parameters and consequences. Having access to the *Hyperdata* of a given agent it is possible to know its functionalities, how they work and how to invoke them. This structure enables the creation of new calls to different types of methods present in different types of agents in run time, without these having been previously implemented during the project. Therefore, with hyperdata, it is possible for a planner to read the data that a problem supplies as input, check all functionalities present in the available agents, and present one or more plans of action.

After recognizing the required set of agents, it is necessary to execute the plan. For that, a new process is open in the monitor agent, which is responsible to invoke, in the correct order, the functionalities of each agent. This process and all the involved agents incorporates the DPA and it is also responsible to check if each action is correctly carried out and in negative cases a new action plan must be calculated.

An important part of this methodology is the planner. It is responsible to provide the action plan and, therefore, must be fast and reliable. Most of the planners present in literature [6-8] are based on intelligent methodologies; however, most of them just provide one possible solution at the end of the search process. By means of a different approach, this paper proposes the utilization of a planner based on the Artificial Immune System (AIS). The AIS is based on the biological principle of bodies' immune systems [9-11]. An immunological system has major characteristics that can be used in search and optimization: proliferation, mutation, selection, and memory. While proliferation is the capability of generating new individuals making the search process dynamic and global, mutation is the ability of searching through the solution space for sub-optimum local points. The selection is responsible for eliminating low-affinity cells, while memory is responsible for storing high-affinity cells from other solutions and using these recollections in new problems intending to reduce the search time. The AIS methodology is based on *niching* process and all the individuals influenced by the same attractor (a local optimum) will converge to a unique point. As result the final population may holds several different ways of solving a problem forming a

repertoire of solutions. This may seem irrelevant, but in real case applications where hundreds of agents are present in the scenario, a planner may take several minutes to generate a solution. Therefore, if a problem occurs during the execution time, it is possible to use this repertoire and check if any other pre-calculated solution fits to the new scenario before calculating a new plan.

This paper is structured as follows: section 2 shows the architecture developed using Aspect-Oriented Modeling, section 3 shows the Artificial Immune System Planner, section 4 shows a simulation using an virtual environment and, finally, section 5 the results are discussed.

2 Using Aspect-Oriented Modeling in Polymorphic Agents

AOP is a programming paradigm that has been proposed to address problems with high index of entanglement among components. It works decomposing the problem into parts for further composition. The great outcome of this paradigm is related with the new composition mechanisms that greatly decrease the number of dependencies among the components. In AOP, problems are decomposed and modeled following the domain knowledge. Some parts of that model compose with the others using OOP mechanisms – those, represent agents, are normal components –, but others require more advanced composition mechanisms – those are called aspectual components of the problem (or aspects, for short). What differentiates an aspect from a regular component is its composition with the rest of the system. A regular component represents an agent of the system and has access only to its internal functionalities. An aspect takes a set of agents present at the model and creates external connections among them. The composition of the system is done through a *Weaver*, an element that has access to all functionalities, and is responsible to bind the different agents together.

The noteworthy property of this methodology is that the agents present on the system are variables themselves – they are not known at programming-time, implying that if any new agent is added to the system, the *Weaver* has automatically access to its metadata without any extra effort. However, to know the metadata contained in the agents present in the system does not implying that the objective of each functionality is known, leading to a typical case where data do not generate information. In order to transform these data into information it is necessary to develop another set of data, called here Hyperdata. Thus, for each method developed in an agent, a *hyperdata* should be developed containing all of the necessary and pertinent information of the functionalities. The information contained in *hyperdata* are related to description, overload, parameters, meaning of the output and parameters, preconditions and effects, but still coherent with the PDDL language, meeting the necessities of the planner, which will be discussed next.

With this set of hyperdata, it is possible to know all functionalities associated with the agents presented in the system with the respective pre-requirements necessary for its execution, as well as the effects obtained with the action execution.

Using the benefits of the AOP methodology, it is possible to develop a highly flexible multi-agent system. The idea is to develop regular components representing each agent (a given device, robot, sensor or even computational system) independent of the problem and scenario that they will work. Each agent has a *Hyperdata* responsible for

informing its actions, variables and predicates. To apply the presented ideas in a multi-agent system, four sets of agents were designed: Execution Agents, Problem Agents, Monitor Agent, and Polymorphic Agent.

Execution Agents (EA) are components that will solve a problem. They represent robots, conveyors, sensors, lifters, computational systems and others. Each one has its own *Hyperdata* responsible to inform how it may be used. The *Problem Agents* (PA) represent PDDL scripts responsible to inform the final desirable state of the system. The *Monitor Agent* (MA) is responsible for seeking all available Execution and Problem Agents present in the system. It is also responsible to read the agents' hyperdata, plan the task list, and weave the used EA along with the Polymorphic agent resulting in the final Dynamic Polymorphic Agent (DPA). The *Polymorphic agent* is responsible to bind the actives EA and execute the tasks necessary to accomplish the final state designed by the PA.

An info-graphic exemplifying these ideas is shown in figure 1 where several Execution and Problem Agents are present in the scenario. The process starts when the Monitor Agent recognizes a specific signature present in the agents. After that it chooses one given problem to solve, runs the planner procedure and generates a task list. This task list is passed to the Polymorphic Agent along with the EAs needed to execute the selected problem. The Polymorphic Agent binds all the agents generating the Intelligent Dynamic Polymorphic Agent (DPA), responsible to fulfill the task. In practical applications, the Polymorphic Agent is a computational system that generates a specific process, the DPA, which is able to invoke the active agents' functionalities. Another important note is that a given Execution Agent may be present in several DPA depending of its availability.

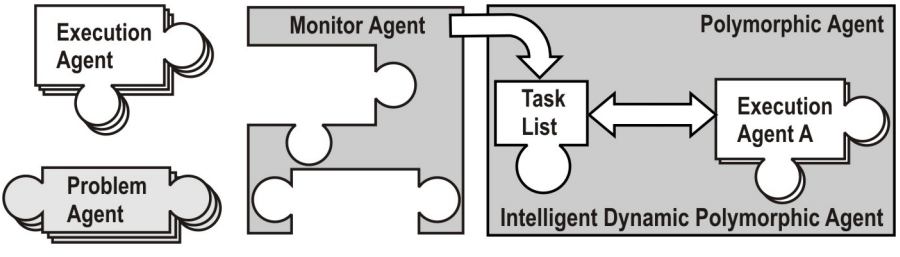


Fig. 1. Info-graphic representing the AOP in Multi-Agents Environments

3 The Artificial Immune System Planner

AI Planning is an area that studies the automatic generation of a plan to solve a problem within a particular domain. Basically, a plan is a sequence of actions provided by a planner that, given an initial state, tries to find how it is possible to achieve some goal conditions. Planners can be domain-dependent or domain-independent. The domain-independent planners are not tied to a particular domain - they can solve problems in a variety of different domains, given a model of that domain in a suitable input language. To standardize the input language, and also, to make easier to evaluate the planners performance, the planning community created the PDDL language,

which is used by most planners, and so, will be used in this work. There are a lot of methodologies used by the planners to reach their goals. Most of them use search methods with heuristic functions. The present proposal is to use a methodology based on the immune system.

The natural immune system (NIS) is responsible for defending the body against dysfunctions from its own cells, and actions from substances and infectious foreign cells, known as non-self elements. The body identifies these non-self elements by using two related systems: the innate immune system and the adaptive immune system. The innate immune system is inborn and unchanging. It ensures resistance to a variety of antigens (Ag's) during their first exposure to the body by providing a set of initial antibodies (Ab's). This general defense operates mostly during the early phase of an immune response. When the body is exposed to a given Ag, the NSI identifies the highest affinity Ab (hAb), and starts the proliferation process. This process is responsible for dividing the hAb, and then generating clones. Many of these clones present somatic mutation from the original cell, generating a new level of affinity to the Ag. The new Ab's, with the highest level of affinity, pass through a process of maturation and become either plasma cells, which are responsible for attacking the Ag's, or memory cells, which store characteristics of the successful Ab's, providing a faster immunological response when, later exposure to the same Ag occurs. An important feature of the NIS is the ability to react against external, harmful agents (non-self or pathogens), while, most of the time, remaining unresponsive to itself (self tolerance). However, the NIS also acts against its own Ab's in a way to benefit itself. In fact, if the number of memory cells increases, the reaction time against a given Ag would also increase, because it should meet every single Ab present in the system before starting the cloning and differentiation process. Therefore, the NIS only memorizes the Ab's with a high affinity level and the other memory cells are eliminated.

The AIS intends to capture some of the principles previously described within a computational framework. The main purpose is to use the successful NIS process for optimization and learning. As in every intelligent-based method, the AIS is a search methodology that uses heuristics to explore only interesting areas in the solution space. However, unlike other intelligent-based methods, it provides tools to perform simultaneous local and global searches. These tools are based on two concepts: hypermutation and receptor edition. While hypermutation is the ability to execute small steps towards a higher affinity Ab leading to local optima, receptor edition provides large steps through the solution space, which may lead into a region where the search for a hAb is more promising.

The technical literature shows several AIS algorithms with some variants. One that has shown good results was the GbCLONALG algorithm presented in [11]. The main statement of GbCLONALG is that progressive adaptive changes can be achieved by using numerical information of the system, instead of only computational brute-force. It will lead to a significant reduction in the number of clones and, consequently, in computing effort. The numerical information to be used can be the *entropy* or just the first order derivatives or gradient, also known as the *tangent vector (TV).*

The GbCLONALG, however, does not use two powerful characteristics present in the NIS: memory cells and maturation control. By using a memory of results from former system states, it may be possible to have a better solution in terms of computational effort and accuracy. It is also possible to assemble a repertoire of solutions for a

given problem, and in case of any marginal change in a system, check if any of the existence solutions is valid. This strategy, besides does not provide new local optima, enables a quick valid solution, avoiding stop the current solution process. The second characteristic is related to the fact that keeping track of the evolution of all artificial Ab's is not a good strategy, because during the hypermutation process several solutions may suffer influence of the same attractor and, therefore, converge to a very similar state. Thus, the identification and further elimination of all similar individuals but the best one would provide a major speed up on the convergence process. This step must be able to correctly identify all attractors present in the system and, for that purpose a cluster strategy will be used.

There are several algorithms for clustering data, and in this work, the MAXMIN distance (MMD) method [12] is used. This method presents two major advantages: it automatically estimates the number of clusters, which it is an essential feature since there is no previously knowledge of the number of local optima present in the system, and it demands only one parameter, which can be heuristically adjusted or can be set by a simple standard deviation method. The algorithm is presented as follows:

To demonstrate these concepts, Fig. 2 shows an example of the niching and cluster process in function (1). Fig. 2a shows the initial population, in 2b the population is at the third interaction and the clusters are defined, and in 2c the maturation control took place and all individual, except the best of each cluster, were eliminated. Finally, Fig. 2d shows the final population over the function mesh. For this particular example, the population started with 60 individuals and, at the end of the simulation process, only 38 remained.

$$Maximize\ \ f(x_1, x_2) = x_1 \cdot sin(4\pi x_1) - x_2 \cdot sin(4\pi x_2 + \pi) + 1. \quad (1)$$

Although the maturation control can provide a major speed-up on the convergence, several optimization problems may present dynamic behaviors that change the original scenario. Even the slightest change implies a full execution of the optimization process, demanding unnecessary computation effort. Thus, the process could still be faster if the initial population presents a high level of affinity, which can be achieved using memory from previous cases. Using this memory from former states of the system, it may be possible to have a better solution in terms of computational response and quality.

Adding these features in the GbCLONALG yields the proposed algorithm shown in Fig. 3, named Cluster-Gradient-based AIS (CGbAIS). Each step or block of this diagram is detailed as follows:

CGbAIS Algorithm:

1. Randomly choose a population $p = \{Ab_1,\ldots,Ab_i,\ldots,Ab_n\}$, with each individual defined as $Ab_i = \{x_1,\ldots,x_j\ldots,x_{nc}\}$, where n_c represents the number of control variables or actions. If there is a memory set present, it must be used as part of the initial population;
2. Calculate the value of the objective function for each individual; this result provides the population affinity for the optimization process;
3. For each individual Ab_i, a new subpopulation of hyper-mutated clones $q_i = \{Ab_{i,1},\ldots,Ab_{i,j}\ldots,Ab_{i,nc}\}$ is generated, where $Ab_{i,j} = \{x_{i,1},\ldots, x_{i,j} + \Delta x_{i,j},\ldots, x_{i,nc}\}$, and n_c

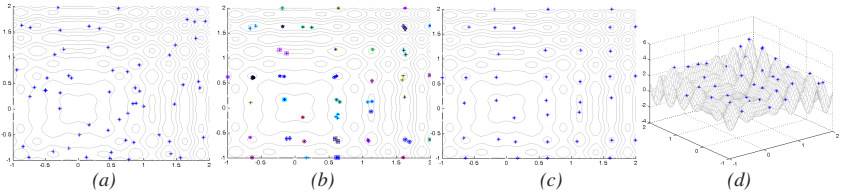


Fig. 2. Niching and maturation control process

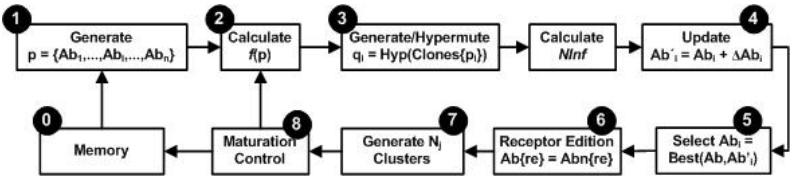


Fig. 3. CGbAIS algorithm

represents the number of hypermutated clones. The hypermutated clones are then used to evaluate the *numerical information NInf* utilized to evolve the population;

4. A new individual $Ab_i^{'}$ is assessed through (2), where ΔAb_i means a value given by the *NInf*,

$$Ab_i^{'} = Ab_i + \Delta Ab_i ; \tag{2}$$

5. Calculate the affinity of this new individual $Ab_i^{'}$ and check if it has a higher affinity compared to the original Ab_i; if it does, the hypermutated clone takes its position in the population p;
6. The bests n_b individuals among the original p population are selected to stay for the next generation. The remaining individuals are replaced by randomly generated new Ab's. This process simulates the receptor edition (re) and helps in searching for better solutions in different areas;
7. Use the MMD algorithm to cluster all individuals that converge to a single attractor. This step will generate the C_j clusters;
8. For each cluster, eliminate all the individuals but the best. If it is the end of the simulation, generate a memory of these individuals.

To adapt the principles of CGbAIS to a dynamic search, some initial considerations must be taken into account:

a) In a discrete scenario, a change in the control variables (in this case represented by using or not a feature) can lead the system into a complete different operational state, making imprecise the idea of small changes around the current state. Thus, during the hypermutation the adopted strategy is to keep the *nLocalBest* clones instead of just one. By using this concept the algorithm allows that a given antibody, which is not well classified at first, may evolve and became the best. This concept is

the same as other successful algorithms such as the *Simulate Annealing* and *Tabu Search*. To avoid a combinatorial explosion, at each interaction, all the clones are compared and just the *nGlobalBest* are kept alive by the maturation control.

b) An antibody represents a partial path over a tree search. It starts at level 1 knowing just one node and, as its evolution occurs new nodes are added to the path. This implies that the initial population can swaps the entire solution space and during the evolutionary process the search starts to be more concentrated in certain areas.

c) In any application, a membership of an element in a given cluster must obey some similarity index. In the case of continuous optimization, the initial population of antibodies evolves and the clusters are found using the MAXMIN distance algorithm. However, in combinatorial optimization, the path in a tree search contains more valuable information than the final result. It indicates the way that an element evolves and can be used as a similarity level. Thus, if an element has the same *n-first* nodes than another one, they can be clustered together at level-n.

d) The clones are responsible for analyzing a given node. Their population expands branches from their parents and the numerical information obtained from this process indicates the likelihood of finding the best solution following a given path. However, the success of this idea depends on the accuracy of the numerical information that presents two major problems. The first one is how to correctly evaluate each branch. If just the minimum value of a clone is taken in account, it implies that all others are disregarded and valuable information may be lost. To avoid this problem, the evaluation function $EvalF_i$ used in the present algorithm is shown as follows.

$$EvalF_i = u_i \times \phi\{ u_1,...,u_{nlocalbest} \} . \tag{3}$$

It takes the mean value ϕ provided by the *nLocalbest* individual of each branch, and multiplies it by each individual clone value u_i. The second problem is that, if a tree search has two or more branches with very similar success likelihoods, the probability of not finding the best one at the end of simulation depends on the proximity of these branches and how many clones were used to generate the necessary numerical information. In scenarios like this, the final solution might be close to the global best and may be easily improved through a local search (hypermutation).

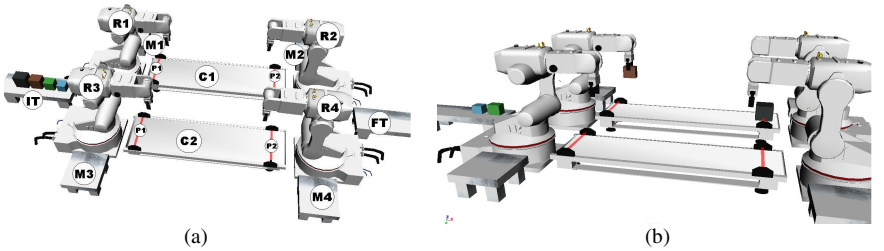
4 Application and Results

In order to validate the methodology, a didactic manufacturing scenario was built using the software VIRTUALMANUFACTURING [13]. The proposed environment is shown in Figure 4. Where there is: two conveyor belts (C1,C2) with positions (P1 and P2), 1 input table (IT), 1 output table (FT), 4 manipulators (R1, R2, R3, R4), 4 storage magazines (M1, M2, M3, M4) and 4 blocks (B1, B2, B3, B4). Under the proposed methodology taxonomy, each of the described items is an Execution Agent. The Monitor Agent, a computational system running in a different computer, uses Remote Procedure Call (RPC) to check and read the Hyperdata of available agents. Thus, each one of the EA must provide Hyperdata containing PDDL scripts and metadata responsible to pass information about how to invoke its functionalities. Back to the virtual scenario, the problem is, given a specific ordering of the input blocks, to

find a sequence of action capable of putting them in the correct order in the output table. The problem supposes that the initial order is (B1, B2, B3, B4) which should be transported to the final table in the order (B4, B3, B2, B1).

The objective passed to the CGbAIS planner – a functionality of the Monitor Agent – is to solve the problem involving the minimum number of execution agents as possible. The output provided 4 possible plans, each one involving different sets of agents, sequences and steps. The processing time was 210ms. Comparing with another planner, the FF achieved 135ms. However, the CGbAIS found 4 possible solutions against just one provided by the FF – the same as the best found by the CGbAIS. An important mention is that, at each action step of each plan, the state of the system is informed. This is crucial if a re-planning is necessary. To solve the problem, the first and most indicated plan is taken in account. This plan has 28 steps and uses as agents R1, R2, R3, C1, M1 and M2. To execute this plan, the MA creates a new process, the Polymorph Agent. The PA binds the agents used in the plan forming one new entity, the DPA, which is responsible to solve the problem. The PA, using the *hyperdata*, is able to invoke the agent's functionalities and therefore, to fulfill the plan.

To simulate a fault in the system, in step 6 the R2 agent was set as unavailable. The DPA detects the fault, informs the MA that reads the current state of the system and check over the 3 other plans if a similar state is present. In this example, the same state is present in solution 3 that uses R1, R2, R4, C1, M1 and M3 to complete the plan and reach the final state. Then, the MA kills the process of the current DPA and starts a new one with this new configuration. An intermediate step of the generated plan can be visualized in Figure 4b.



(a) (b)

Fig. 4. (a) Application Environment using the VIRTUALMANUFACTURING (b) Intermediate step

5 Conclusion

This paper proposed an innovative architecture of intelligent agents. Using a framework that works through aspect-orientation, it is possible to generate a methodology based on hyperdata capable of describing and invoking the functionalities of the agents. A Monitor Agent is responsible for identifying available functionalities, planning the actions and generating a polymorphic entity during run time, dedicated to the resolution of a determined problem. The methodology uses open concepts of domain-free modeling, which provides a high-level of reutilization. A new planner, based on Artificial Immune System, is able to provides several local optima and, keeping memory of these alternatives solutions, it is possible to check for new solutions if a

fault happens. This major advantage avoids the necessity of searching for a new plan. To test the method, a virtual scenario using the software VIRTUALMANUFACTURING was used, however, not only manufacturing problems are applicable to the system. Any domain that can be modeled in the PDDL language, can be inserted in the system scenario. More complex and real domains are the goal in this work's next step. The results demonstrated that the approach was able to find a plan and quickly re-plan the actions and involved agents under the presence of a fault. Therefore the system has proven to be extremely flexible and with great applicability power.

References

1. Tabuada, P., Pappas, G.J., Lima, P.: Motion Feasibility of Multi-Agent Formations. IEEE Transactions on Robotics 21(3) (2005)
2. Bagnall, A.J., Smith, G.D.: A multiagent Model of the UK Market in Electricity Generation. IEEE Transactions on Evolutionary Computation 9(5) (2005)
3. Honorio, L.M., Barbosa, D.A., Souza, A.C.Z., Lopes, C.V.: Intelligent optimal power flow system development using aspect-oriented modeling. IEEE Trans. on Power Systems 22(4), 1826–1834 (2007)
4. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V.: Aspect-Oriented Programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241. Springer, Heidelberg (1997)
5. Russel, K., Norvig, P.: Artificial Intelligence, A Modern Aproach. In: Planning, ch. 11, pp. 337–366. Prentice Hall, Englewood Cliffs
6. Hoffmann, J.: FF: The Fast-Forward Planning System. AI Magazine 22(3), 57–62 (2001)
7. Do, M.B., Khambhampati, S.: Solving planning-graph by compiling it into csp. In: Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling (2000)
8. Gerevini, A., Serina, I.: LPG: a Planner based on Local Search for Planning Graphs with Action Costs. In: Proceedings of the 6th International Conference on Artificial Intelligence Planning Systems, AIPS 2002, pp. 13–22 (2002)
9. Castro, L.N., Zubben, F.J.V.: Learning and optimization using the clonal selection principle. IEEE Trans. on Evolutionary Computation 6(3), 239–251 (2002)
10. Honorio, L.M., Leite da Silva, A.M., Barbosa, D.A.: A gradient-based artificial immune system applied to optimal power flow problems. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 1–12. Springer, Heidelberg (2007)
11. Xu, L., Chow, M.-Y., Timmis, J., Taylor, L.S.: distribution outage cause identification with imbalanced data using artificial immune recognition system (AIRS) algorithm. IEEE Transactions on Power Systems 22(1), 198–204 (2007)
12. Friedman, M., Kandel, A.: Introduction to pattern recognition: statistical, structural, neural and fuzzy logic approaches. World Scientific Publishing, London (2000)
13. Honorio, L.M., Dias, W., Freire, M., Souza, L.E.: Virtual Manufacturing System, Program and Video Tutorials (in Portuguese), http://www.virtualmanufacturing.unifei.edu.br (Project CNPq/CT-Info 400842/2003-3) (online since 2006)

AIS-Based Bootstrapping of Bayesian Networks for Identifying Protein Energy Route

Sungwon Jung, Kyu-il Cho, and Doheon Lee

Department of Bio and Brain Engineering
Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
{swjung,kicho,dhlee}@biosoft.kaist.ac.kr

Abstract. It is important to identify the mechanism of energy transfer in protein structures in understanding their functions. Highly enriched energy in some hot spots of protein structures is transferred to other residues during some functional activity such as binding. The transferred energy reaches at various residues and make them to change their three dimensional structures to make other functional effects. In this paper, we use Bayesian network learning in identifying the route of energy transfer from the estimated energy status of residues. Artificial immune systems (AIS) approach is used for bootstrapping the Bayesian network learning. The analyzed results give a quantitative map of route for energy transfer in 1be9 protein.

Keywords: Protein energy transfer, Bayesian network, Hot spot, Artificial immune systems.

1 Introduction

Proteins are basic building blocks of biological functions in living organisms. There are many approaches in revealing the functions of specific proteins. Among various perspectives, the structural characteristics of proteins are considered as major clues in revealing their functional roles.

In general, genes exist in living cells as sequences of nucleic acids (e.g., DNA or RNA). For each of specific functional roles, some selected sequences are transcribed and finally compose an amino acid sequence according to the original sequence of the nucleic acids. The amino acid sequence then folds into some three dimensional structure according to its chemical characteristics, energy status and electromagnetic force between those amino acids.

Fig. 1 shows a three dimensional structure of the protein 1be9. Such a three dimensional structure plays a significant role in interacting with other proteins or molecules. When a protein binds other proteins or molecules, their three dimensional structure should match the structural shape of the binding position. Once binding occurs successfully, the energy levels of the residues in the binding positions become higher than before. These highly activated energy is transferred to other residues in the protein according to its structural characteristics. The

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 176–187, 2008.

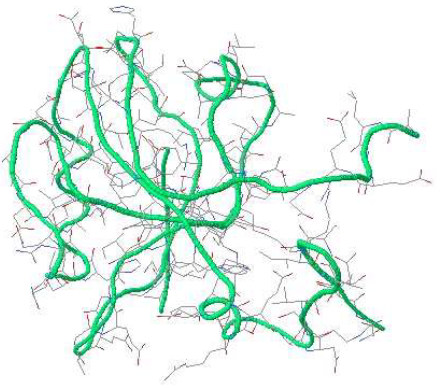


Fig. 1. Three dimensional structure of the protein 1be9. From the Protein Data Bank (http://www.rcsb.org).

transferred energy then reaches some residues and can change their structural characteristics. Thus understanding the mechanism of energy transfer in proteins is very important in understanding their actual functional mechanisms.

There have been various approaches to identify the mechanisms of energy transfer in protein structures [1] [2]. Popular studies are researches on finding out binding sites of proteins. Not only the structural shapes of those binding sites but also energy levels have been significant interests of many researchers.

The initial approach for energy levels in protein structures is identifying the quantitative energy level of energy-enriched residues in binding sites. Such quantitative energy levels were measured by experimental approaches and also expected from various indirect information such as amino acid sequences [1] [2]. Because it is not easy to quantitatively measure such binding energy levels via experiments, it has been common approach to expect the energy level of residues from amino acid sequence conservation information [1]. The intuition in this approach is that highly conserved amino acid sequences may play important roles in functioning of the specific protein. Thus it is widely accepted that such highly conserved regions of protein sequences are possible candidates of binding sites and can have highly enriched energy levels. Several studies were conducted to expect the energy levels of protein residues during binding processes. Such approaches mainly target to identify the specific energy levels and binding sites which are usually called *hot spots*.

Another issue related to protein energy is understanding the changing dynamics of energy status in protein structures. Chennubhotla and Bahar [2] used a Markov propagation model [3] [4] [5] of energy transfer in protein structures to expect the distribution of energy levels after energy transfer dynamics arrived at the stable status in a protein structure. The Markov propagation model was successful to some extent in expecting the energy status after binding of proteins. However, the Markov transition matrix, which describes the energy diffusion process from each residue, is not proper for identifying possible route

of energy transfer. For this reason, we propose a method to identify the route of energy transfer, which was initially defined with the Markov transition matrix. A Bayesian network model which describes the dependencies of energy transfer is built using energy transfer instances from the Markov propagation model. An artificial immune systems-based bootstrapping approach is used to construct sub-optimal Bayesian networks to give quantitative information on the route of energy transfer.

We first present in Sect. 2 the previous Markov propagation model of energy transfer in a protein structure. Then, Sect. 3 mentions the approach of using Bayesian networks to describe the route of energy transfer. We also explain how the Bayesian network model can be a compact description of energy transfer which was originally defined as a Markov transition matrix. In Sect. 4 we present artificial immune systems-based bootstrapping approach for learning a number of sub-optimal Bayesian networks. The learned results on the route of energy transfer are described in Sect. 5. Lastly, we conclude and give some perspectives in Sect. 6.

2 Markov Propagation Model of Energy Transfer

In this section, we introduce the previous Markov propagation model of energy transfer in protein structures. In Chennubhotla and Bahar's approach, they used a discrete-time, discrete-state Markov process model for energy transfer between residue pairs. The Markov process model sets the transfer probability between residues. The probability of energy transfer between two residues r_i and r_j can be defined as their interaction strength, also called affinity. First, the (i, j) element of the affinity matrix A can be defined as

$$a_{ij} = \frac{N_{ij}}{\sqrt{N_i N_j}} \tag{1}$$

where N_{ij} is the number of atom-atom contacts between two residues r_i and r_j within a given specific cutoff physical distance between centers of residues. N_i and N_j are the number of heavy atoms in the corresponding residues r_i and r_j. This definition comes from the intuition that more contacts between atoms have larger affinity in composing structures. The affinity of self contact $a_i i$ can be defined from this intuition, but we do not consider the self affinity here.

Using this definition of affinity, the local interaction density d_j of a residue r_j can be defined as

$$d_j = \sum_{i=1}^{n} a_{ij} = \sum_{j=1}^{n} a_{ji} \tag{2}$$

where n is the number of whole residues in the protein. If we consider d_j as a diagonal element of the diagonal matrix $\mathbf{d}$,

$$\mathbf{d} = \text{diag}\{d_j\} \tag{3}$$

we can define the conditional probability m_{ij} of transferring energy from residue r_j to residue r_i in one time step as follows:

$$m_{ij} = \frac{a_{ij}}{d_j} \tag{4}$$

Because d_j works as a normalizing factor, the sum of every conditional probability of energy transfer becomes 1 at residue r_j.

$$\sum_{i=1}^{n} m_{ij} = 1 \tag{5}$$

We can define the conditional probability matrix $\mathbf{M} = \{m_{ij}\}$, also called the Markov transition matrix as follows:

$$\mathbf{M} = \mathbf{A}\mathbf{d}^{-1} \tag{6}$$

Now let us denote the initial energy level at some residue r_j is p_j^0. If we use the conditional probability of energy transfer m_{ij}, the amount of energy transferred from r_j to r_i is $m_{ij}p_j^0$. Assuming linear summation of such transferred energy to r_i from every residue, we can use the following matrix notation of energy distribution after one time step:

$$\mathbf{p}^1 = \mathbf{M}\mathbf{p}^0 \tag{7}$$

where $\mathbf{p}^k = [p_1^k, ..., p_n^k]$. This propagation process can be iterated. After β steps, the distribution of energy on residues can be represented as:

$$\mathbf{p}^\beta = \mathbf{M}^\beta \mathbf{p}^0 \tag{8}$$

As $\beta \to \infty$, $\mathbf{p}^\beta$ converges to a stationary distribution, given by $\pi_i = d_i / \sum_{k=1}^{n} d_k$. Chennubhotla and Bahar analyzed these stationary distribution for their interested proteins and their results showed patterns of energy distribution in protein structures in the binding processes. Their results showed that such a method for identifying energy distribution pattern can be predictive in finding out energetically related residues.

The Markov transition matrix represents information on how the energy is transferred between residues in a protein. It is a proper approach when we want to see the global patterns of energy distribution on the residues. However, we need to make an abstracted energy transfer route for finding out residues which are significantly involved in the whole energy transfer process.

3 Using Bayesian Network Learning for Route Identification

Previous studies [2] showed patterns of energy distribution in protein structures. The Markov propagation model can describe local characteristics of energy

transfer from each residue. However, the Markov transition matrix used in the Markov propagation model may not be proper in identifying the route of energy transfer in the protein structure. Basically, the transfer of energy from any residue in the structure can be directed all directions toward other residues. Even though such entire information of energy flow can be important, the abstraction of the energy flow in the protein structure can be very helpful in compactly describing and understanding such dynamic flow of energy in the protein structure.

The Bayesian network model is the graphical representation of a joint conditional probability distribution [6]. A Bayesian network B is represented as (G, Θ), where G is a directed acyclic graph and Θ is the set of probabilistic parameters. Each node in G corresponds to a probabilistic random variable and each edge represents conditional dependency between two random variables. Representing the whole joint probability distribution of a set of random variables is infeasible. By assuming Markov condition, where a random variable is independent of its nondescendents given its parents, a Bayesian network can represent the target joint conditional probability distribution in a very compact way.

There are many ways in learning an optimal Bayesian network B from observed data $\mathbf{D}$ [8] [9] [11] [12] [10] [7]. Because B is composed of G and Θ, learning Bayesian networks is composed of two parts - learning the dependency structure G and learning the probabilistic parameters Θ. Learning Θ is easier than learning G in general because once we have some dependency structure G, then we can find out corresponding probability values to the dependency structure from the observed data $\mathbf{D}$. Further, our interests is much more in learning the structure G because the dependency structure between the residues in a protein can represent possible *major* route of energy transfer. Learning the structure G from the given observed data $\mathbf{D}$ is to find out G which maximizes $P(G|\mathbf{D})$. This can be considered as a conventional search problem for optimal solution and several scoring schemes have been proposed.

In this paper, we focus on the ability of Bayesian networks where some probabilistic distribution can be compactly represented in the graphical form. For applying the Bayesian network model to our problem, we need to consider each residue in a protein as a probabilistic random variable which has some probability values for possible values of energy level. Let us assume that each residue r_i is a probabilistic random variable where specific probability value exists for each possible energy status of r_i. One instance of observation for the set of random variables $\mathbf{R} = \{r_1, r_2, ..., r_n\}$ now corresponds to one observation of energy distribution for those residues, and we denote it as x_k. If we observe the different energy distribution on the residues in a protein m times, an observed set of data instances $\mathbf{D} = \{x_1, x_2, ..., x_m\}$ can be constructed. By learning an optimal Bayesian network $B = (G, \Theta)$ from $\mathbf{D}$, we can get a graphical representation of conditional dependencies between protein residues. Because the conditional dependencies in a learned Bayesian network represent the probabilistic dependencies between energy levels of different residues, we can interpret the dependency structure G of B as a route of energy transfer of the protein.

We should observe the distribution of energy level in a protein structure experimentally to build the observed data instances $\mathbf{D}$. However, experimental observation of energy level for certain protein can be very difficult. Instead, we use the Markov propagation model to generate a set of observations on the energy distribution. Suppose that we have a set of s initial energy distribution $\mathbf{P}^0 = \{\mathbf{p}_1^0, \mathbf{p}_2^0, ..., \mathbf{p}_s^0\}$. For each $\mathbf{p}_k^0$, we can get $\mathbf{p}_k^1$, $\mathbf{p}_k^2$, ..., $\mathbf{p}_k^\beta$ by iteratively multiplying the Markov transition matrix $\mathbf{M}$ until convergence. Then we can make $\mathbf{D}$ as follows:

$$\mathbf{D} = \{\mathbf{p}_1^1, ..., \mathbf{p}_1^\alpha, \mathbf{p}_2^1, ..., \mathbf{p}_2^\beta, ..., \mathbf{p}_s^1, ..., \mathbf{p}_s^\gamma\} \tag{9}$$

By the way, the number of observed data instances $|\mathbf{D}|$ is usually insufficient to learn an optimal Bayesian network. The number of residues in our target protein 1be9 is 115. Learning an optimal Bayesian network for 115 random variables may require tremendous amount of observations. Note that more than 30,000 observations were used to learn Bayesian networks of 37 random variables with around 95% of structural accuracy in previous studies. Because searching an optimal directed acyclic graph (G of the optimal B) is an NP-hard complexity, the required number of observation data grows super exponentially for 115 random variables. For this reason, we use artificial immune systems-based bootstrapping approach for learning several sub-optimal Bayesian networks. The approach of learning Bayesian networks using artificial immune systems-based bootstrapping will be discussed in the following section.

4 AIS-Based Bootstrapping of Bayesian Networks

In this section, we present our AIS-based bootstrapping approach for learning several sub-optimal Bayesian networks. We use the clonal selection algorithm. The benefit of using clonal selection algorithm for Bayesian network analysis is that learning Bayesian network structures and the bootstrapping procedure is handled simultaneously. In conventional bootstrapping methods, any selected learning algorithm should be used for each subset of given training data. Because learning Bayesian network structures is a very complex problem (NP-hard), greedy algorithms are widely used in for the learning method in the case of bootstrapping where not just one but $k(>> 1)$ structures should be learned. The use of clonal selection algorithm for Bayesian network structure bootstrapping provides population-based learning algorithm for subsets of given training data. Thus incorporating the clonal selection algorithm for bootstrapping let us avoid local minimum with more chances and capture more true patterns of the solution space. The conventional structure of the clonal selection algorithms is described in Algorithm 1.

For our case, an antibody ab_j corresponds to a Bayesian network structure G_j. Antigens are constructed from the observed energy distributions $\mathbf{D} = \{\mathbf{p}_1^1, ..., \mathbf{p}_1^\alpha, \mathbf{p}_2^1, ..., \mathbf{p}_2^\beta, ..., \mathbf{p}_s^1, ..., \mathbf{p}_s^\gamma\}$. We define an antigen ag_i as a subset of $\mathbf{D}$ and make every ag_i has the same size. As mentioned in the previous section,

Algorithm 1. ClonalSelection

```
1: Randomly initialize a population of n antibodies Ab
2:
3: while Stopping criteria do
4:    for each antigen ag_i in Ag do
5:       Determine affinity to each antibody ab_j in Ab
6:       Select m(< n) antibodies of highest affinity from Ab
7:       Clone and mutate the selected antibodies
8:    end for
9:
10:   Select l(< n) antibodies of highest affinity from Ab to make new Ab'
11:   Add random n − l antibodies to Ab'
12:   Replace Ab with Ab'
13: end while
```

D is constructed using the Markov propagation model. Then each antigen ag_i is generated by randomly selecting the predetermined number of instances from **D**. The affinity between an antibody ab_j and an antigen ag_i should represent how well the antibody(G_j) matches to the antigen(a subset of observed data instances **D**). Thus we can use one of existing scoring schemes for Bayesian network structures given data instances as an affinity measure, such as BDeu score G_j given ag_i [13]. The BDeu score evaluates $P(G|\mathbf{D})$ for a graph structure G given observed set of data instances **D**. Mutation of an antibody ab_j(a Bayesian network structure G_j) can be done by using one of following structure modification operations - edge addition, edge deletion and edge reversion. By replacing corresponding terms and operations in Algorithm 1 with those for Bayesian networks given above, we can define an algorithm *ClonalBootstrapping* for learning several sub-optimal Bayesian network structures as follows:

5 Results and Analysis

5.1 Environments

We analyze the route of energy transfer in the structure of protein 1be9. 1be9 protein is one of the PDZ domain family. There are total 115 residues (from r_{301} to r_{415}. Indexes are given from number 301.) in 1be9. 15 residues among those 115 are selected as *hot spots* by conservation score analysis, which are energetically enriched during the binding processes. The conservation score analysis evaluates the degree of conservation during evolution for a given gene sequence by comparing gene sequences of many different kinds of species. Selected 15 residues have higher evolutionary conservation scores than other residues. Higher evolutionary conservation score means that the sequence of the residue have been conserved more during the evolution process of species. This supports a widely accepted hypothesis that well conserved regions of a protein do important roles in the biological processes. By interpreting the conservation score

Algorithm 2. ClonalBootstrapping

```
Randomly initialize a population of n graph structures Ab

while There is improvement larger than θ in average BDeu(G_j|ag_i) do
   for each antigen ag_i in Ag do
      Evaluate BDeu(G_j|ag_i) for each antibody G_j in Ab
      Select m(< n) graph structures of highest score from Ab
      Clone and mutate the selected structures
   end for

   Select l(< n) structures of highest score from Ab to make new Ab'
   Add random n − l structures to Ab'
   Replace Ab with Ab'
end while

Return k structures of highest score from Ab
```

of a residue as a degree of being a *hot spot* and thus as an enriched amount energy for that residue, we assigned following initial energy level to 15 residues in Table 1.

Table 1. Initial energy levels of *hot spot* residues

Residue	Energy (kT^*)
r_{318}	0.3
r_{322}	0.2
r_{323}	2.3
r_{324}	2.7
r_{325}	6.4
r_{326}	2.2
r_{327}	1.5
r_{328}	2.0
r_{329}	0.5
r_{331}	2.0
r_{339}	0.5
r_{372}	5.7
r_{376}	2.0
r_{379}	2.3
r_{380}	1.9

When 1be9 binds to another protein or molecule, every *hot spot* residue may not be bind to its counterpart. For example, binding proteins to 1be9 may have different mutated sequences in some part of their binding residues and thus some of those 15 *hot spots* may not bind to their counterparts. Then the energy enrichment may occur only subset of those 15 *hot spots* where the counterpart protein

binds successfully. This means that there are $2^{15} = 32768$ sets of combination of the initial energy distribution for those 15 residues. We constructed $\mathbf{P}^0$ with 1,820 $\mathbf{p}_k^0$s by randomly selecting 1,820 combinations of initial energy distribution among 2^{15} combinations for 15 residues and setting the energy level of other 100 residues as 0. By iteratively multiplying the pre-computed Markov transition matrix $\mathbf{M}$ to each of 1,820 $\mathbf{p}_k^0$s, $\mathbf{D}$ of 32,760 instances of energy distribution for protein 1be9 is finally constructed.

The *ClonalBootstrapping* of Algorithm 2 was applied to $\mathbf{D}$ for learning $k = 100$ suboptimal Bayesian network structures. The number of antibodies in the population was 1,000, the number of antigens was 100, $m = 200$ antibodies were selected during the processes, and $l = 950$ antibodies were survived at each epoch.

As a result, we have got 100 suboptimal Bayesian network structures. A confidence value was evaluated for each possible connection between two residues r_i and r_j in the structure as follows:

$$Conf(r_i \rightarrow r_j) = \frac{\sum_{G_s \text{ including } r_i \rightarrow r_j} BDeu(G_s|\mathbf{D})}{\sum_{\text{All } 100\ G} BDeu(G|\mathbf{D})} \tag{10}$$

With enough convergence, the 100 suboptimal structures have very similar scores each other while they can have different topologies. Thus the confidence value for an edge can be approximated in easier way as follows:

$$Conf(r_i \rightarrow r_j) \approx \frac{\text{The number of } G_s \text{ including an edge } r_i \rightarrow r_j}{100} \tag{11}$$

5.2 Results

Figure 2 shows the result of our analysis. Nodes correspond to residues and edges correspond to conditional dependencies between residues. Those conditional dependencies can be interpreted as major routes of energy transfer between residues. Edges with confidence less than 0.5 (edges which were shown less than half of 100) were omitted. Nodes with no connected edges of confidence larger than 0.5 were omitted also in the figure. Gray colored nodes are 15 *hot spot* residues in Table 1. The thickness of edges linearly corresponds to the confidence value, where confidence of 0.5 corresponds to the thickness of 0.5pt and confidence of 1 corresponds to the thickness of 12pt.

From this result, we can identify in which route the energy is transferred between residues. Even though the direction of edges in Bayesian networks do not mean causality, it is known that the order in the Bayesian networks has some correlation with causality. Thus we can consider the direction of edges as a direction of energy flow to some extent. This can be useful because we can find the destination residues of energy flow from selected starting residues, such as binding residues.

In Figure 2, we can find that the map of route is disconnected into several sub-networks. This may represent that there may exist *several different valleys*

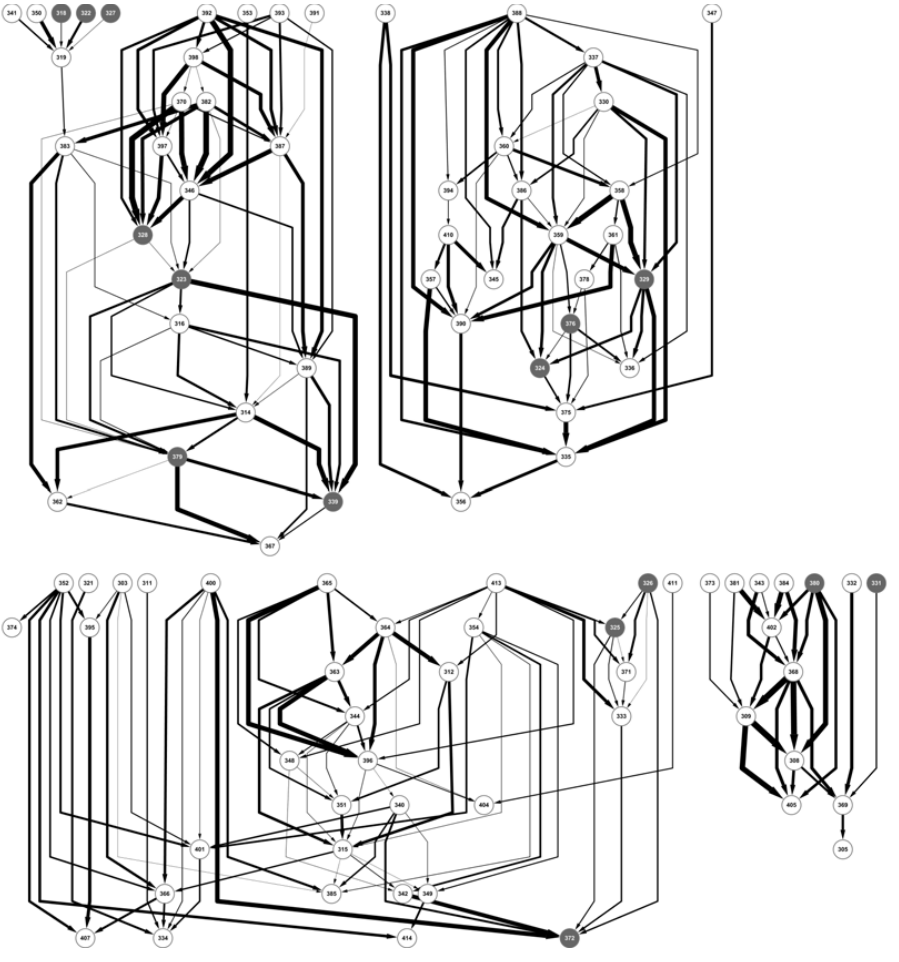


Fig. 2. Estimated route of energy transfer in protein 1be9

of energy flow even in a single protein. Even though there is little experimental evidence which can support this hypothesis, this analysis of identifying energy route can give some clue about it. Identifying several valleys of energy transfer in a protein can give a blue print for modifying the protein structure with our preferences. Suppose that there is a starting residue and a target residue of energy transfer in a protein. The energy transfer may alter the structural shape or molecular status of the target residue. If we want to block the energy transfer to the target residue, we may need to change the structure of some residues included in the energy transfer. Now assume that there are two energy transfer routes between starting and target residues. If it is hard to change the structures of residues in one of the routes, we can take another as our experimental target

route. This is just an example with a hypothesis on the energy route but can give intuition on the impact of our method.

6 Conclusion

In this paper, we have presented the AIS-based bootstrapping of learning Bayesian networks. The proposed approach has been applied to the identification of energy transfer route in a 1be9 protein. The result can be interpreted as a map of major energy transfer route between residues.

A bootstrapped learning algorithm of Bayesian networks has been proposed based on the clonal selection algorithm. The proposed algorithm incorporates several schemes which are needed in learning Bayesian networks. A lot of analysis problems suffer the lack of data situation. Besides the conventional bootstrapping and Monte Carlo methods, this clonal selection-based bootstrapping can be another choice. We have learned the Bayesian network model in this paper, but that approach can be adapted to any other models where bootstrapping can be applied.

Further works may include the experimental or theoretical validation of the learned energy transfer route in the protein. The relationships between initial residues of energy transfer and the destination residues needed to be studied. During the works given in this paper, we did not set those *hot spot* residues to become the topmost nodes in the route. If we have some knowledge on the initial location of energy diffusion or the order of the energy transfer, we can adapt that information in maintaining the population of antibodies. Addition of such a prior knowledge can improve the quality of this route analysis.

The comparison of the presented AIS-based bootstrapping method was not compared to conventional bootstrapping methods. Further, the power of learning Bayesian networks with the proposed method can be compared with Markov Chain Monte Carlo method for learning Bayesian networks. This future work will show the actual characteristics, pros and cons of the clonal selection-based bootstrapping of Bayesian network learning.

References

1. Lockless, S.W., Ranganathan, R.: Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families. Science 286, 295–299 (1999)
2. Chennubhotla, C., Bahar, I.: Markov Propagation of Allosteric Effects in Biomolecular Systems: Application to GroEL-GroES. Molecular Systems Biology 4, 1–13 (2006)
3. Kullback, S.: Information Theory and Statistics. Dover Publications, New York (1959)
4. McLachlan, G.J., Basford, K.E.: Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York (1988)
5. Chennubhotla, C., Jepson, A.: Hierarcical Eigensolver for Transition Matrices in Spectral Methods. Adv. in Neural Information Processing Systems 17, 273–280 (2005)

6. Neapolitan, R.E.: Learning Bayesian Networks. Pearson Prentice Hall, London (2004)
7. Acid, S., Campos, L.M.: BENEDICT: An Algorithm for Learning Probabilistic Belief Networks. In: 6th International Conference IPMU, Granada (1996)
8. Brown, L.E., Tsamardinos, I., Aliferis, C.F.: A Novel Algorithm for Scalable and Accurate Bayesian Network Learning. In: MEDINFO (2004)
9. Cooper, G.F., Herskovits, E.: A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning 9, 309–347 (1992)
10. Etxeberria, R., Larrañaga, P., Picaza, J.M.: Analysis of the Behaviour of Genetic Algorithms when Learning Bayesian Network Structure from Data. Pattern Recognition Letters 18, 1269–1273 (1997)
11. Friedman, N., Nachman, I., Peer, D.: Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm. In: Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 206–215 (1999)
12. Jung, S.W., Lee, K.H., Lee, D.H.: Enabling Large-Scale Bayesian Network Learning by Preserving Intercluster Directionality. IEICE Trans. Inf. & Syst. E90-D, 1018–1027 (2007)
13. Heckerman, D., Gerger, D., Chickering, D.M.: Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. Machine Learning 20, 197–243 (1995)

A Neuro-Immune Inspired Robust Real Time Visual Tracking System

Yang Liu[1], Jon Timmis[1,2], and Tim Clarke[1]

[1] Department of Electronics, University of York, UK
{yl520,jt517,tc}@ohm.york.ac.uk
[2] Department of Computer Science, University of York, UK

Abstract. We present a novel Neuro-Immune inspired real-time tracking system that is capable of tracking morphing moving targets over non-benign backgrounds. We have employed ideas from antigen-presenting cells, T-cell interaction, together with cytokine interaction with neural systems. Our experiments show that the neuro-immune tracking system has the ability to maintain tracking a target even if the target changes shape, or is covered for periods of time by other objects.

Keywords: Neuro-Immune inspired, Visual tracking, Morphing target, Non-benign background, Cellular Immune Network (CIN).

1 Introduction

This paper proposes a real-time visual tracking system that is capable of tracking objects whose aspect, shape and/or size change whilst they transit across a background that is likely to confuse the tracking process due to a transient similarity to the object being tracked. We define this type of deformable object a *morphing* target and we define the background against which it moves as *non-benign*. Real time tracking is an important subject in machine vision applications, and accuracy, robustness and speed are the primary concerns for a reliable real time tracker. There are two major approaches in designing a visual tracker, *model-driven* and *data-driven*.

The model-driven approach, also known as a top-down process, has its roots in control theory. Visual information is abstracted into a state space and tracked by applying Bayesian filtering techniques. The essence of Bayesian filtering theory is a recursive process of prediction and correction, given *a priori* knowledge and proofs. With this knowledge, we could clearly (though not easily) track the target and outline its profile in the image even if the background is cluttered. Many ad-hoc Bayesian filters for visual tracking have been developed, such as Extended Kalman Filters [1]. Unfortunately, many problems occur when applying these filters to practical applications. Firstly, there may be limited *a priori* knowledge about the object, i.e. the exact motion model and morphing laws. Secondly, abstracting the feature vector from a deformable target is computationally expensive, since deformable targets suffer from a more complicated representation compared to rigidly-shaped objects. Two alternative solutions are

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 188–199, 2008.

either to enhance conventional computing capabilities and structures to satisfy such algorithms, or, alter the algorithms to fit current hardware conditions many of which are still slow and cumbersome.

Data-driven approaches are generally bottom-up processes, and derive from the area of image processing. It is often preferable to use the actual image data from the camera rather than an estimation from a Bayesian filter. Some simple visual tracking systems, not exploiting filtering techniques, turn to image segmentations. There are many image segmentation algorithms: some e.g. [2] are simple histogram based, edge detection and region growing; and others are more sophisticated like level-set [3]. Compared with model-driven methods, data-driven ones do not need a predefined model, and significantly, they are much faster especially when supported by a dedicated pixel processing hardware platform such as a Field Programmable Gate Array (FPGA). However, data-driven methods are not so robust at dealing with a cluttered environment. The advantages and shortcomings of the two approaches are mutually exclusive that many researchers [4] in the machine vision community often attempt to combine them and make them operate complementarily. Motivated by these characteristics, in this paper, a novel visual tracking method is introduced, which is inspired by biological, neuro-immune interactions. It has features of both model-driven and data-driven approaches, and exploits their individual advantages to provide robust and fast tracking of morphing targets against non-benign backgrounds.

The paper is structured as follows: Section 2 introduces a novel immune-inspired 2-D planar array platform ready for image segmentation, Section 3 presents the neuro-immune inspired tracking model, Section 4 analyse the system performance when applied to a representative test scenario and Section 5 summarises the tracking system.

2 A Cellular Immune Network Platform

In this section we outline a novel image segmentation approach, based on ideas taken from the immune system, that is capable of visual representing a object of interest in the image and track morphing, moving targets over non-begin backgrounds.

2.1 The Structure and Representation of CIN

In order to quickly and robustly represent an object of interest, a system should provide a reliable image segmentation ability and parallel computation features. 2-D cellular computing platforms are widely used in image processing literature due to their 2-D planar array structure. On these platforms, a single cellular unit corresponds to a pixel in the image. The image processing function on a pixel level is equal to the state updating rule of the cellular units. There are many examples of applications that have provided good image processing results of applying such a structure, for example work using cellular automata (CA) [5], cellular neural network (CNN) [6] and pulse coupled neural network (PCNN)

[7]. However in these examples, algorithms on CAs are typically employed for noise filtering and algorithms on CNN are typically employed for edge detecting; they are not used in general for object representation. Although PCNN performs well at image segmentation, the output is a series of pulse-codes which is not convenient to present to an object before an extra frequency processing module. What is required is a new mechanism dedicated for simple image segmentation, i.e. distinguishing the object of interest from the background. In this section, a cellular immune network (CIN) platform is proposed to visually represent the target. The final output of CIN is binarised (1 and 0) to indicate the membership of a pixel being part of the target or not.

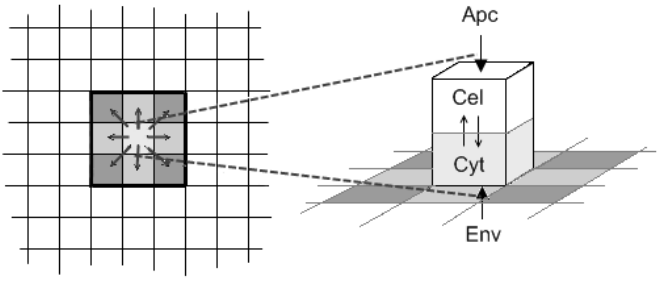


Fig. 1. The structure of CIN

In our CIN, T-cells are classified by the types of cytokines they produce. We define each type of T-cell as being able to secrete several kinds of cytokines, with different types of T-cells being able to secrete the same kinds of cytokines. However, in our case each type of T-cell is affected only by a single type of cytokine. Therefore, the number of cytokine types that a certain type of T-cells is able to secrete is equal to the number of T-cells types that this concerned type is able to act on. We map all types of T-cells into the CIN planar array. Figure 1 illustrate the 2-D array structure of CIN. T-cells and the kind of cytokines exclusively acting on it are coupled and represented by a cellular unit. This mapping mechanism means that the distance between 2 cellular units does not represent the physical distance between two T-cells, but represents the difference between two types of T-cells according to their cytokine secretion ability i.e. the type of cytokines they secrete.

In the context of image processing, each pixel is represented by such a cellular unit with each unit having two non-negative state variables Cel and Cyt, representing a T-cell concentration and the coupled cytokines concentration, with respect to the planar position (i, j). For instance, $Cel(i, j)$=0 means the concentration of T-cells of type (i, j) are zero. Seg is defined as the CIN output which is the binarised Cyt value by a threshold (see figure 2, block "activation threshold") and represents the segmentation result: 1 means part of the object

and 0 means that it is not part of the object. Each cellular unit has two inputs *Env* and *Apc*. *Env* is the environmental factor which is defined as the inverse of gradient value at each pixel. This value is always less than 1 and designed to decrease the cytokine concentration at various areas. *Apc* has 3 states: absent, inactivated and activated. Any unit with *Seg* value of 1, is with the *Apc* state of absent; otherwise, it is inactivated if did not find the SIBO pixels (took as antigens), and if found, the *Apc* state turns to be activated. This mechanism is for anti-interference in visual tracking.

2.2 The Interactions of the Cellular Units

In our CIN, the closer the 2 cellular units are, the more the same kinds of cytokines they are able to secrete, which means they have similar cytokine secretion ability. We establish the rule that each kind of T-cells secretes certain kinds of cytokines only to affect the units in it's neighbourhood (including itself) and define the cytokine secretion ability of each type of T-cells as:

$$M(x,y) = exp(-\frac{x^2+y^2}{r^2}) \qquad x,y \in [-n,n] \tag{1}$$

Where (x,y) is the relative position according to the concerned cellular unit and r is a nonlinear scaling parameter of the amplitude of M, and therefore the cytokine secretion ability decreases as the distance increases. n is the radius of the cytokine secretion area, and reflects the cytokine secretion ability. If it is mostly one, then this means that the cytokine secretion area for a cellular unit is in the closest neighbourhood, including it's eight closest neighbours and itself. So $M(x,y)$ is also expressed as a 3x3 matrix, as figure 2, block "Secretion matrix" shows. In the figure, colour represents the value of coefficient in M: the brighter the unit's colour, the higher the coefficient value.

At each pixel, e.g. at position (i,j), the Cyt value is defined as equation 2. This means at each grid the cytokines concentration is the summation of all amount of cytokines secreted from the neighbourhood. Since an image usually is considered as a 2-D matrix, this equation is also expressed in a 2-D convolution form by equation 3, where $Conv2(,)$ is the 2-D convolution function, operator '$\bullet$' is the element-by-element multiplication of two matrices. In this form, cytokine secretion matrix M is the convolution kernel. Equation 2 and 3 are mathematically equivalent.

$$Cyt(i,j) = Env(i,j) * \sum_{x=-n}^{n} \sum_{y=-n}^{n} (Cel(i+x, i+y) * M(x,y)) \tag{2}$$

$$Cyt = Env \ \bullet \ Conv2(Cel, M) \tag{3}$$

For any time step, the Cel value is updated by equation 4 and 5. The proliferation rate ΔCel in a cellular unit is decided by a proliferation function $f(\cdot)$ with respect to its corresponding Cyt value. The bell-shaped function of $f(\cdot)$ is

shown in figure 2, by the block "proliferation function". Looking at the function diagram, there are two limits value along the Cyt axis, a lower limit and a higher limit. Where the Cyt value is below the lower limit or over the upper limit, ΔCel value is zero. t is the time step, k is the natural death rate, and c is the proliferation rate which mimics the process of T-cells encountering antigen presenting cells (APC) in an activated state. These input states of Apc is for anti-interference in visual tracking system and will be discussed later in section 3.1.

$$\Delta Cel(t) = f(\ Cyt(t)\ ) \tag{4}$$

$$Cel(t+1) = \Delta Cel(t) + Cel(t) - k + c \tag{5}$$

2.3 The Image Segmentation Process

Our image segmentation algorithm can be likened to a diffusion process. The Cel and Cyt are all initialised at zero, and then an area is selected as a seed area (a small group of conjunctive pixels) in the object area, we set their Cel and the system begins operation. First of all at their close neighbour pixels, the Cel value will increase from zero, if their Cyt value at a moderate level. Over a certain threshold Cyt, the pixel in question is considered as part of the object.

As the recognised area expands, the environment parameters act as a stoping force to inhibit it at the object edge area. Edges are generally the pixels with high intensity gradient value. Intensity gradient is the local intensity changes between conjunctive pixels and usually calculated by edge detectors [2]. As previously mentioned, the inverse of gradient value at each pixel works as the environment parameter, we know from equation 2 that a very small value of environment parameter decreases, or dilutes, the cytokine's concentration, and the proliferation function shows that a very low cytokine concentration inhibits the cell's proliferation. Therefore, the intensity gradient value could limit the T-cells expanding across the edge. After several iterations, the output should cover the whole visual object in the image and maintain the shape the same as the object's.

This section has introduced the immune-inspired image processing platform CIN. Segmenting the object of interest is the preliminary work for a visual tracking system. The whole tracking model is to be discussed in the next section, where the CIN platform is employed by a a neuro-immune inspired visual tracker.

3 Neuro-Immune Inspired Tracking Method

The immune and nervous systems were considered to be two independent systems until the second half of 20th century. Since then, biologists have found significant evidence to show that these two systems interact at many levels using a variety of signalling materials [8]. From an immune system perspective, T-cells secrete different kinds of cytokines which regulate the immune response

and affect neural behaviours. From the neural perspective neurons secrete hormones, neuropeptides and neurotransmitters to regulate the immune response.

Such interactions between the two systems did not attract a great deal of interests from computer scientists or engineers until [9]. In that paper, the authors describe each function of the immune, nervous and endocrine system and their interactions and propose an artificial homeostasis system (AHS) which attempts to maintain the system's internal stability. However, in terms of implementation the authors concentrate on the interactions of the neural and endocrine system: that work has in part inspired work presented in this paper. Other influential work for our system was proposed in [10] where the authors develop a simple innate artificial immune system integrated with a self organising map (neural network). In our work, we make use of standard feedforward neural networks and have taken inspiration from the adaptive (rather than the innate) component of the immune system.

In this section we outline a novel tracking system inspired by immune and nervous systems interactions. The immune-inspired system is to visually present a object of interest using CIN, and the neural-inspired system is to track the target and dynamically learn the hidden motion rules.

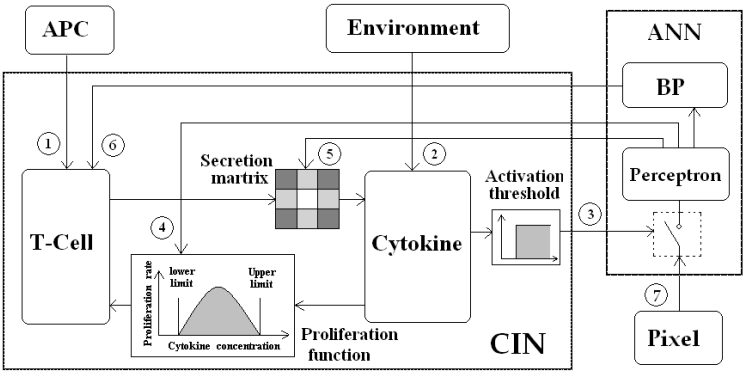


Fig. 2. Neuro-immune inspired tracking system framework

3.1 Immune-Neural Framework

Our framework is illustrated in Figure 2, and is composed of a CIN component and a ANN component in dashed box. The ANN is a combination of an artificial perceptron and a back propagation (BP) network. In figure 2 blocks represent system elements and arrows show functions with one element acting on another. Within the CIN block, there are elements responsible for T-cell concentration *Cel* of a cellular unit: current T-cells concentration, T-cell proliferation function $f(\cdot)$ and cytokine secretion matrix M, environment parameters input *Env*, and high proliferation when *Apc* is activated. For instance, the *T-cell* block convolving by the cytokine *secretion matrix* acts on the *cytokine* block and backwards

is affected by through the T-cell *proliferation function*. These are graphical representations of equation 2, 4, and 5.

In our tracking system there are four functional modules for robust segmentation, fast segmentation, adaptive tracking and anti-interference, respectively. The four modules are divided by their specific functions and each is composed of several elements and functions.

Robust segmentation. In this module, the neural-inspired global information analysis mechanism helps the immune-inspired local pixel processing platform, CIN, to iteratively segment an object from the background. This is the first and most important module in the visual tracking system. This module is composed of 4 actions represented by arrows 2, 3, 4 and 7.

Arrow 2, from block "environment" to block "cytokine", represents the effect of environment parameters Env onto the Cyt values in equation 2. Env is defined as the inverse of gradient value at each pixel. However the gradient information is not always sufficient. If the image's contrast is very low, the gradient values will be very small and thus the environment parameters will not be small enough to inhibit the recognised area.

With further consideration, it is known that the gradient information is the local information, that only depends on a few continuous pixels' intensities. Besides the local information, analysing the global information from the whole image is more helpful. The pixels' intensities are the inputs of the perceptron (see figure 2 arrow 7), so the size and number of the perceptron's inputs are as the same as the image's resolution and also as the units of the CIN. Once the Seg value of a CIN unit is 1, i.e. the Cyt value on a pixel is above the threshold, the corresponding neural synapse is activated and will pass the pixel's intensity value to the perceptron (see figure 2 arrow 3). At every iteration, the perceptron collects the intensities and positions information from all pixels in the "cytokines activated" area, and calculates the intensity and position distribution, e.g. the mean value and the standard deviation. The outputs of the perceptron are the membership probabilities of these concerned pixels, as equation 6 defined,

$$Pr(i,j) = \exp(-\frac{(I(i,j) - \mu_t)^2}{2{\sigma_t}^2}) \tag{6}$$

where $Pr(,)$ is the perceptron output to the concerned pixel, $I(,)$ is the intensity value of the concerned pixel's; μ_t and σ_t represent the current mean value and standard deviation respectively. This is the effect as figure 2 arrow 4 shows, from the perceptron output to act on the proliferation fucntion. The new proliferation function is expressed as equation 7, and the proliferation rate in equation 4 will be recalculated by replacing $f(\cdot)$ with $f_{new}(\cdot)$.

$$f_{new}(\cdot) = f(\cdot) * Pr(i,j) \tag{7}$$

Equation 6 and 7 show that if a pixel with lower membership probability value, even if the gradient value is small, the Cyt value could also be very small. Thus this pixel will be considered not being part of the object. With the feedback from perceptron unit, the CIN is able to right segment an image with low contrast.

Fast segmentation. This module is concerned with accelerating the segmentation process via modulating the cytokine secretion matrix M by the perceptron output, as figure 2 arrow 5 indicates. The secretion matrix M represents the T-cell's cytokine secretion ability and is symmetric (see equation 1). Since the recognised area is changing, in most cases enlarged, the mean value of the position distribution, which is the centre position of the recognised area, is also changing. If the centre position is moving, this means that along a certain direction the recognised area is "growing" faster than along others. Therefore, by biasing the matrix with the centre's moving speed, the cytokine secretion ability will be stronger along the centre's moving direction. Therefore the cells expansion will be anisotropic, faster along some directions and slower along some others. This will provide faster segmentation ability.

The centre position of currently recognised area is defined as P_t which is composed of x and y coordinates. Along the iterative segmentation process, P_t is moving towards the actual centre of the object, and will finally reach and stop there once the recognised area covers the whole object. The error of the centre positions in two continuous iterations is considered as the moving speed V_t. It is defined as equation 8, where (t) is current iteration time step.

$$V_t(t) = P_t(t) - P_t(t-1) \tag{8}$$

V_t and P_t are different from V_f and P_f respectively, which are the speed and centre position of the a frame rather than a iteration, and will be introduced latter. The biased secretion matrix $M_{new}(,)$ is defined as equation 9, where $M_v(,)$ is the bias matrix describing the moving centre speed and as the same size as M.

$$M_{new}(x,y) = M(x,y) + M_V(x,y) \tag{9}$$

Having a biased cytokine secretion matrix affords the CIN with a faster segmentation ability. This is also provided by the perceptron output. The artificial perceptron provides the membership probability to affect the proliferation ratio of each cellular unit, and the centre of recognised area to affect the cytokine secretion matrix.

Adaptive tracking. In a reliable visual system, the observation from the image is usually used instead of the estimation from a Bayesian filter since the error between them is very small. Although the motion model of the tracking target is unknown, the model could be updated by the error of previous prediction and current observation (EPO) defined as equation 10,

$$EPO(f) = V_f(f) + V_f^*(f) \tag{10}$$

In our real time tracking system, a BP neural network is introduced as the time series predictor, in a similar vein to [11]. The network input is the observed speed $V_f(f)$ in the current frame and output is the predicted speed $V_f^*(f+1)$ for the next frame, where (f) is the sequential number of current frame. Use the EPO to regulate the neural synapse weights on the fly, figure 3. In a discrete system,

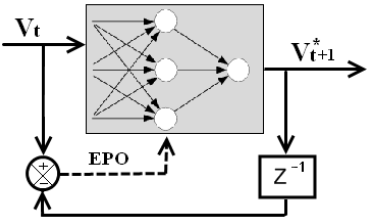


Fig. 3. BP network as a time series predictor

the term "speed" is actually the position difference between two continuous frame, so after each frame processed, the BP network shifts all the T-cells to the predicted position $P_f^*(f+1)$. This is the function of figure 2 arrow 6 represents. The predicted position is defined as equation 11. $P_f(f)$ is the current position.

$$P_f^*(f+1) = P_f(f) + V_f^*(f+1) \tag{11}$$

Anti-interference. After the BP network shifted the recognised area to the predicted position, the next video frame is taken. The CIN will perform the segmentation process again for the new frame, and self-adapt to the new shape of the morphing target. If the target encounters a SIBO, the tracker may l be confused and take the SIBO as part of the target. In the CIN platform, if there is no discrimination mechanism, the recognised area will keep expanding into the SIBO. In order to avoid such a situation we have developed an anti-interference mechanism.

Equation 4 indicates that when the *Apc* state is activated at the concerned cellular unit, the *Cel* value is raised to a very high level. This is also the effect that figure 2 arrow 1 indicates. From the proliferation function, it shows that not only could low cytokine concentration inhibit T-cell's proliferation and the aggregation's expansion, the high cytokine concentration would have the same effect. This feature provides good anti-interference potential to restrain the recognised area from leaking into the SIBO area. The *Apc* state only has local impact on the target-SIBO-overlapped area, and on any other areas recognised area are free to expand. That means the tracker is able to track a morphing target and well define its contour even when it's passing through a SIBO. The function of *Apc* state is like a lever to balance the predication and observation.

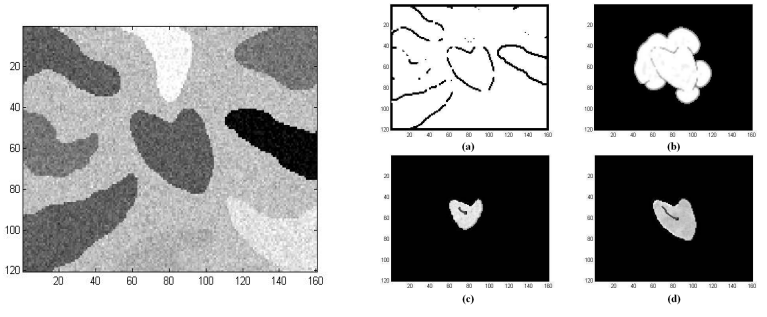
4 Performance Analysis

This algorithm has been simulated and tested in the MatLab 2006a environment and is available from the authors on request. It consists of two independent tests, for image segmentation and visual tracking. All testing images are 256 gray scales, by 160x120 pixels and applied Gaussian noises on. For space, we report only a single example run of the system, but extensive experiments have been undertaken but are not reported here.

4.1 Image Segmentation

In this test, we focus on the system's robust and fast segmentation abilities. Figure 4(a) is a raw image. The heart-shaped object in the middle is the target to be segmented. Picture (a), is the gradient of the raw image and (b) shows the failed segmentation when using the CIN in its own; and (c), (d) are the segmentation results (after 50 iterations) when using the CIN *with* an artificial perceptron, where the left is slower and the right is faster.

Figure 4(b) (b) shows that without the help of artificial perceptron, the CIN image segmentation fails: the recognised area "leaks" outwards from the gaps (lower gradient value area) in the contour. Picture (a) is the corresponding edge detecting result, which can't provide a continuous contour of the object. The successful segmentation is shown in picture (d), and when compared with picture (b), we deduce that our system provides significant robustness.



(a) The raw image (b) Segmentation Results

Fig. 4. Visual Representation of the Segmentation Process

It is necessary to clarify that the algorithm applied in picture (c) is also as robust as the algorithm in picture (c), only slower. The difference between two algorithms employed in picture (c) and (d) is that the faster of the two employs the biased matrix mechanism, whilst the slower algorithm does not. Given the same time length, in this experiment 50 iterations, we can see in picture (c) the work has is approximately half completed (without leaking), while in picture (d), the process is complete. The starting point is at the top-left of the object, the centre of recognised area is moving towards the bottom-right corner along with recognised area expanding. The dark thine line in the object is the trajectory of the moving centre.

4.2 Visual Tracking

In this experiment we are testing the visual tracking ability. Figure 5, column (a) is a series of frames from a simulated video clip at number 0,5,10,15,20 and 25, where a morphing target is passing through a SIBO. In column (b) we observe

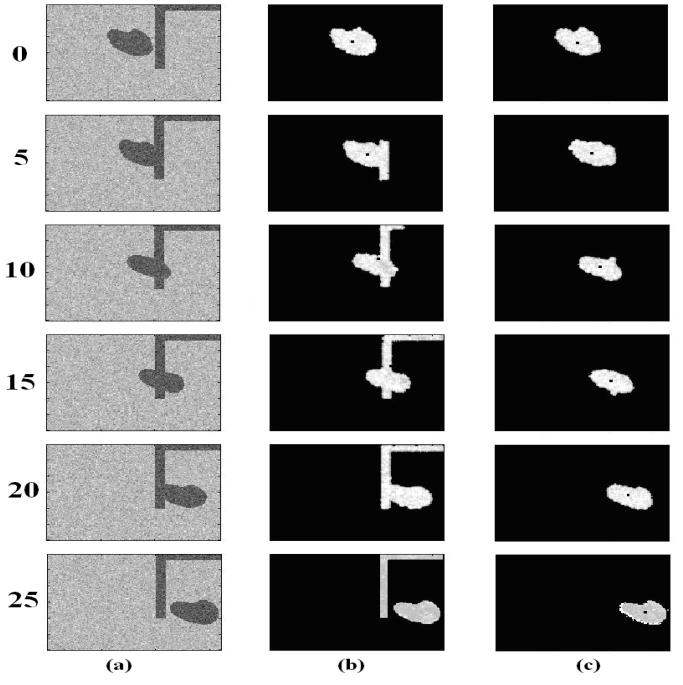


Fig. 5. Images taken over time showing the tracking of a morphing target

that without the immune-inspired anti-interference mechanism, the recognised area expands into the SIBO. The black dot in the recognised target represents the centre coordinates. We see the centre of our target is surely aberrant like being attracted by the SIBO, and after frame 15 it is out of the object, which implies that the target has been lost. However, in column (c), although there is some distortion in the SIBO overlapped region, the algorithm is able to maintain the centre position inside the target area. The slight distortion is a compromise between the prediction and observation. When the algorithm is tracking normally, the algorithm relies on the observation of the actual pixel information from the image, but once the SIBO appears, this tracker relies on the prediction more than the current observation.

5 Summary and Conclusion

In this paper we have proposed a novel visual tracking system which is inspired by neuro-immune interaction. After identifying a seed point in the object of interest in the first image frame, the system will segment the object from the background and will successfully track the object despite the fact that the object is changing its aspect and passing through a cluttered background. There are two tightly coupled decision making mechanisms for visual tracking: the CIN

makes decisions on which pixel belongs to the target object, and the ANN makes decisions on where the recognised area is moving. The two systems are pixel-driven and model-driven respectively. Both of them mutually act on each other via their outputs, to regulate the behaviours of each other's. Benefited by the two features of pixel-driven and model-driven, this system provides robust and fast tracking abilities.

References

1. Strom, J., Jebara, T., Basu, S., Pentland, A.: Real time tracking and modeling of faces: An ekf-based analysis by synthesis approach, Corfu, Greece (1999)
2. Jain, R.C., Kasturi, R., Schunck, B.G.: Machine Vision. McGraw-Hill, New York (1995)
3. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: A level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(2), 158–175 (1995)
4. Oliver, N.M., Rosario, B., Pentland, A.P.: A bayesian computer vision system for modeling human interactions. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 831–843 (2000)
5. Rosin, P.L.: Training cellular automata for image processing. IEEE Transactions on Image Processing 15, 2076–2087 (2006)
6. Murguia, M.I.C.: Texture segmentation by the 64x64 cnn chip. In: 7th Int. WS on Cellular Neural Networks and their Application: Nonlinear Information Processing and Intelligent Sensors, pp. 547–554 (2002)
7. Murguia, M.I.C., Zimmerman, A.S., Pablo Rivas, P.: Image processing applications with a pcnn. Advances in Neural Networks, 884–893 (2007)
8. de Castro, L.N., Timmis, J.: Artificial Immune systems: A new Computation Intelligence Approach. Springer, Heidelberg (2002)
9. Timmis, J., Neal, M.: Artificial homeostasis: Integrating biologically inspired computing. Technical Report UWA-DCS-03-043, University of Wales, Aberystwyth (February 2003)
10. Neal, M.: Don't touch me, i'm fine: Robot autonomy using an artificial innate immune system. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 349–361. Springer, Heidelberg (2006)
11. Wan, E.A., Nelson, A.T.: Neural dual extended kalman filtering: Applications in speech enhancement and monaural blind signal separation. In: IEEE Workshop on Neural Networks for Signal Processing, pp. 466–475 (1997)

Negative Selection with Antigen Feedback in Intrusion Detection

Wanli Ma, Dat Tran, and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, Australia
{Wanli.Ma,Dat.Tran,Dharmendra.Sharma}@canberra.edu.au

Abstract. One of the major challenges for negative selection is to efficiently generate effective detectors. The experiment in the past shows that random generation fails to generate useful detectors within acceptable time duration. In this paper, we propose an antigen feedback mechanism for generating the detectors. For an unmatched antigen, we make a copy of the antigen and treat it the same as a newly randomly generated antibody: it goes through the same maturing process and is subject to elimination due to self matching. If it survives and is then activated by more antigens, it becomes a legitimate detector. Our experiment demonstrates that the antigen feedback mechanism provides an efficient way to generate enough effective detectors within a very short period of time. With the antigen feedback mechanism, we achieved 95.21% detection rate on attack strings, with 4.79% false negative rate, and 99.21% detection rate on normal strings, 0.79% false positive. In this paper, we also introduce Arisytis – Artificial Immune System Tool Kits – a project we are undertaking for not only our own experiment, but also the research communities in the same area to avoid the waste on repeatedly developing similar software. Arisytis is available on the public domain. Finally, we also discuss the effectiveness of the r-continuous bits match and its impact on data presentation.

Keywords: Artificial Immune System, Negative Selection, Intrusion Detection System.

1 Introduction

Artificial Immune System (AIS) is a branch of computational intelligence, inspired by biologic immune systems. It was first proposed by Forrest et al [1] and has attracted increasing interest from the research communities in the last 20 years [2-5]. Like the other biologically inspired models, such as Artificial Neural Networks, Evolution Algorithms, and Ant Colony etc., AIS is based on the observation of the behaviors and the interaction of antibodies and antigens in a biological system [6, 7]. Negative selection, clonal selection, and immune network theory are the three most popular theories of the current AIS research [3, 8].

Negative selection [9-11] mimics the way a human body detects and destroys harmful antigens. A human body constantly produces lymphocytes, with randomly mutated surface peptides, from born marrow. A lymphocyte is recognized by its

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 200–209, 2008.

surface peptides, because these peptides are used to match against other cells. All newly generated lymphocytes are sent to thymus to mature. The thymus has almost all types and shapes of self cells. During this period of maturing time, if a lymphocyte matches any cell in the thymus, the lymphocyte is just a copy of a self cell and is then destroyed. Only these which do not match any self cell in the thymus are sent to the body to match, or detect, antigens (also called pathogens), which are invasion cells. The lymphocytes keep trying to match all the cells, including both self cells and antigens, in the body. If a match happens, it basically means that a non-self cell (antigen) is just detected. An alarm might be raised, and immune actions may follow. The lymphocyte which matches the antigen may become a memory lymphocyte and stays in the body to quickly respond to the same invasion in the future. If for a period of time, a lymphocyte does not make any match, it will age and die. New lymphocytes with random peptide mutations are being generated to replace the dead ones. For the detailed explanation on how the immune system works, under the context of AIS, we refer the readers to [12, Chapter 2].

The terms used in AIS literature are yet to be standardized. In this paper, we use the terms *antibody* and *detector* interchangeably. We also use a *memory antibody* (or detector) to mean that the antibody has been successfully activated and also matched the incoming antigens many times. Finally, we view the data to be verified, i.e., to be matched by the antibodies, as a stream of *antigens*. For the purpose of simplicity, we call all data items to be verified as *antigens*, regardless of being attacks or self.

Negative selection, due to its ability of discriminating self and non-self, fits naturally into the area of intrusion detection. There are a few proposals of using negative selection for intrusion detection purpose. The first, and perhaps the most cited system, is LISYS by Hofmeyr etc. [7, 11]. In [13], Balthrop reported comprehensive results of different parameter settings of LISYS. In LISYS, detectors and antigens are represented as the strings of 49 bits long. Gabrielli and Rigodanzo [14] proposed a similar intrusion detection system but restricted their experiment on the HTTP requests to a web server. Gonzalez and Dasgupta proposed a real-valued negative selection (RNS) algorithm [15], where detectors and antigens are represented as real valued vectors. They tested the algorithm on MIT Lincoln Lab DARPA 99 dataset and achieved 95%-98% detection rates (with different false alarm rates). Ji and Dasgupta had a comprehensive survey paper on the development in negative selection [16].

The success of negative selection depends on the success of generating detectors. In [17], Kim and Bentley reported the difficulties in generating useful detectors within an acceptable time window. They concluded that negative selection suffers from scalability problem. Our experiment also repeated their observations; however, we do not share their conclusion. The problem is not on negative selection itself, and the solution is on finding a means to efficiently generate effective detectors.

In this paper, we propose an antigen feedback mechanism to efficiently generate effective detectors. In addition to the randomly generated detectors, for an unmatched antigen, we copy it into the detector space and treat it the same as a randomly generated detector. The detector is called a *feedback detector* (or a *feedback antibody*). This new detector goes through the same maturing process and is subject to elimination if it matches any of the self strings. If it survives, it is used to match further incoming antigens. If it can be activated, by exceeding the pre-set activation threshold, it becomes a legitimate detector.

The antigen feedback mechanism is justified. The goal of randomly generating detectors is not the randomization, but to generate effective detectors. The ideal situation is that every randomly generated detector matches a type of incoming antigens. Therefore, there is no waste on the generated detectors. This is the ideal situation, which has maximum efficiency, but unachievable in reality. Keeping the goal of randomly generating detectors in mind, it is acceptable to copy an unmatched antigen into the detector space, as it is the same as any one of these many randomly generated detectors. However, this simple antigen feedback mechanism has great impact on the quality of detectors. It makes the detector generating almost close to maximum efficiency.

We employ a similar implementation as LISYS but use KDD CUP 1999 dataset [18] to conduct our experiment on the antigen feedback mechanism. The experiment is conducted on an in-house developed system called Arisytis. Arisytis (Artificial Immune System Tool Kits) is a project we are undertaking in the attempt to implement the up to date AIS algorithms. It does not just provide a test bed for our own research, but is also available in the public domain for the other research communities to avoid the waste of time in repeatedly developing similar software (http:// staff.ise.canberra.edu.au/dtran/).

In this paper, we will also discuss the impact of *r-contiguous bits match* method [7, 11] on the presentation format of antibodies and antigens.

The rest of the paper is as follows. In Section 2, we briefly introduce Arisytis. Section 3 provides the details of preparing KDD CUP 1999 dataset for our experiment. Section 4 discusses the generating of detectors. Section 5 gives the experiment results, with the discussion on the r-contiguous bits match method and its impact on data presentation. We conclude the paper with future work in Section 6.

2 Arisytis

Arisytis is a project we are currently undertaking, through which we are trying to integrate the up-to-date AIS algorithms into a single program. Arisytis is designed for research experimenting and educational purposes. It is highly configurable and also provides real time updates on its run-time activities. The real-time updates make it less efficient, but it is justified for its purposes. Changing Arisytis parameters is just a matter of filling forms and ticking boxes. Arisytis is developed on Microsoft C#.NET environment and has an intuitive graphic user interface, Fig 1.

The top region of Arisytis window has 4 groups: *Global Parameters*, *Antibodies*, 4 *buttons* (OpenTrain, OpenTest, Run/Rerun, and Quit), and *Running Environment*. The elements in Global Parameters group are used to change system wide parameters, such as, the number of allowed antibodies, effective length of an antibody, the duration of a clock cycle, and the time to live for a newly generated antibody etc. The elements in Antibodies group have the parameters related to antibodies, for example, the value R for r-continuous bits match and if antigen feedback is turned on etc. The 4 buttons are used to open relevant files and start an experiment run. The elements in Running Environment group are updated in real time manner to report system activities.

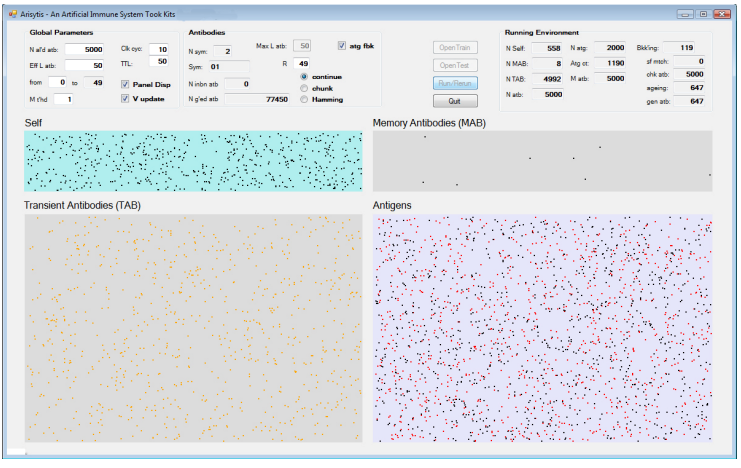


Fig. 1. Arisytis user interface

The lower region of Arisytis window has four panels to display the self strings, antibodies, and antigens. *Self* panel displays self strings, one dot per string. *Transient Antibodies Panel* contains generated antibodies, one dot per antibody. A green dot means that the corresponding antibody is newly generated and hasn't gone through the maturing process yet. An orange dot means that the corresponding antibody survives the maturing process and is ready to match against incoming antigens. After the number of the matches exceeds the predefined matching threshold, which can be adjusted in Global Parameters group, this antibody is promoted into a memory antibody and thus moved to *Memory Antibody* panel. *Antigens* panel displays all the antigens, one dot per antigen, as usual. Although we view antigens as a flow of antigens coming from outside, for displaying purpose, we display all antigens in this panel without the order. An orange dot means that the corresponding antigen has been matched by a memory antibody, i.e., is recognized or detected. A black dot means that the corresponding antigen is either not recognizable by the memory antibodies or yet to be processed. While the display of the antigens ignores the order of the antigens in the antigen flow, the program processes the antigens one by one in order through its flow. The order of the antigens being processed is important, especially when the antigen feedback mechanism is turned on.

3 Preparing the Dataset

KDD CUP 1999 dataset was based on MIT Lincoln Lab intrusion detection dataset, also known as DARPA dataset [19]. The data was produced for "*The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining*" [18]. The raw network traffic records have already been converted into vector format. Each vector has 41 fields (features). We refer the readers to

[18] and [20] for the meanings of these fields. If we ignore the fields with symbolic values, the rest of the fields can be classified into 4 categories:

- *Group I*: fields 0, 4, 5, and 7, these fields are the basic characteristics of a connection. They are the durations, the octets transferred, and wrong fragmentation flags of the connection. We ignore the fields with symbolic values, field 1, 2, 3, and 6, in this paper.
- *Group II*: fields 10-19, these fields are actually not traffic features. The values cannot be obtained by looking at the traffic records alone. The help from host based logs is needed.
- *Group III*: fields 22-30, these fields are time based traffic features. They are the statistics of traffic features in the previous 2 second time window. The calculation is based on the source IP address.
- *Group IV*: fields 31-40: the same as Group III, except that the calculation is destination IP address oriented.

Among the 4 groups, either Group III or Group IV contributes most to the detection rate, and combining the groups won't increase the detection rate [21]. Therefore, we primarily choose Group III fields for our experiment. In addition, we also choose Field 1, 2, and 3. We convert these fields into a string of 50 bits, which has 10 segments as follows:

- *Segment 1*: 1 bit for Field 24. The value range for the field is 0-0.94 with predominately 0s. If the value is 0, we set the segment 0; otherwise 1.
- *Segment 2*: 1 bit for Field 25. The value range for the field is 0-1 with predominately 0s. If the value is 0, we set the segment 0; otherwise 1.
- *Segment 3*: 9 bits for Field 22. The value range for the field is 1-511. We convert the value into its binary format.
- *Segment 4*: 9 bits for Field 23, the same as Segment 3.
- *Segment 5*: 3 bits for Field 1. There are only 3 different values for this field: TCP, UDP, and ICMP; therefore, 100 for TCP, 010 for UDP, and 001 for ICMP.
- *Segment 6*: 7 bits for Field 2. There are 70 different services for this field, for example, auth, ftp, http, and telnet etc. We order the 70 services, in alphabet order, from 1 to 70, e.g. auth ordered as No. 2, ftp No. 17, http No. 22, and telnet No. 59. We then convert the order value into its binary format.
- *Segment 7*: 11 bits for Fields 3. There are 11 distinct flags for the fields. They are OTH, REJ, RSTO, RSTOS0, RSTR, S0, S1, S2, S3, SF, and SH. We set 1 bit of the 11 bits to 1 for one of the flags, the same as Segment 5.
- *Segment 8*: 1 bit for Filed 28. The value range for the field is 0-1 with almost all values as 1s. If the value is 1, we set the segment 1; otherwise 0.
- *Segment 9*: 1 bit for Field 29. The value range for the field is 0-1 with almost all values as 0s. If the value is 0, we set the segment 0; otherwise 1.
- *Segment 10*: 7 bits for Field 30. The value range for the field is 0-1 with even distribution. We first times the value with 100 (range: 0-100) and then convert the product into its binary format.

We use the file "kddcup.data" to generate self strings. We pick up all the vectors with the label "normal" and then convert them into the bit strings as described. The file contains 972,781 vectors with the "normal" label. After being converted into bit strings, we obtain 23,587 unique strings. With these 23,587 unique strings, most of them are just a one off string, and only 558 strings have each individual string being repeated for more than 100 times. We keep these 558 strings as self strings.

We use the file "corrected" to generate the testing strings. The file contains 311,029 vectors, among which 60,593 are labeled as "normal", and the rest 250,436 are labeled with verities of attacks. For our experiment purpose, we pick up the first 10,000 of "normal" labeled vectors and attack labeled vectors respectively and convert them into bit strings.

4 Generating Detectors

The efficiency and the accuracy of the detection are decided by how well the detectors are generated. We encountered the same difficulty as report by Kim and Bentley [17] – we cannot generate even a single useful detector in a period of time which is even far beyond the time required to process all antigens. A simple analysis reveals that the difficulty is actually expected.

We use a 50 bits string to represent an antibody or an antigen. It means that the number of all possible strings is 2^{50}, which is about 10^{15}. As discussed before, from the 250,436 vectors with attack labels, we only obtain 12,351 unique strings. If we generate the detectors completely randomly, and assume that the random numbers are evenly distributed, the chance for us to generate a useful detector is $\frac{12,351}{10^{15}} \approx 10^{-11}$, which is fundamentally impossible.

To overcome the problem, we introduce the antigen feedback mechanism. For any unmatched antigen, we copy it into the antibody repository. It is then treated the same as a randomly generated detector and is subject to the same maturing, eliminating, and activating processes. If it survives, it becomes a legitimate detector.

5 Experiment Results and Discussions

With the data described in Section 3 and the antigen feedback mechanism, we run a number of experiments on the 10,000 attack strings and 10,000 normal strings, with the different combinations of R for r-continuous bits match and T for the number of matches to activate an antibody. Table 1 lists some sample results. In the table, R is the value of r-continuous bits match, T is the activation threshold, "D Rate" means detection rate, "M ATB" lists the number of memory antibodies obtained during this run.

For the attack strings, we have 3 types of results. When R is small ($R \leq 32$), the results are random, as the match is too general to produce any meaningful results. The larger the value R is, the more specific the match is. When $R = l$, where l is the full length of a string, any two strings have to be exactly the same to make a match; while

Table 1. Some sample results of different combinations of R and T

R	T	Attacks		Normal	
		D Rate	M ATB	D Rate	M ATB
49	3	94.96%	8	98.50%	3
	5	93.95%	4	98.92%	2
	10	90.67%	3	99.45%	1
40	3	94.99%	8	98.51%	2
	5	93.99%	4	98.81%	2
	10	90.43%	3	99.60%	1
32	3	65.65%	41	98.41%	5
	5	65.59%	37	98.73%	4
	10	61.31%	25	99.60%	1

at the other extreme $R = 0$ means that any two strings will always match. When *R* is close to the full string length ($R \geq 35$), the detection rates are around 94%, when the activation matching threshold was set at 5. The error rates 6% (from the detection rate 94%) were mainly caused by the initial learning process. The system has to learn, from the feedback antigens, and then activated the detectors. The higher the activation matching threshold we set, the higher the error rate. If we keep the memory detectors of the last run and insert them into the next run, we can achieve almost 100% detection rate. The insertion of memory detectors can be viewed as immunization injection. When $R = 33$ or $R = 34$, we achieved the best results, 95.19% and 95.21%, respectively, when the activation threshold was set at 5. In these 2 cases, *R* is large enough to avoid mistakenly matching the self strings and also is not yet specific enough (i.e., even larger) to exclude similar attacking strings. For all the experiments we conducted, the useful detectors were actually all obtained from antigen feedback, and none of them were generated randomly.

For normal strings, when R is small ($R \leq 31$), we achieved 100% detection rate persistently. This is understandable. The randomly generated detectors rarely match the incoming antigens, while any feedback antibody always matches the self strings and thus is eliminated. Therefore, no match against the incoming antigens can be made. When R is large enough ($R \geq 32$), the match against self strings becomes more specific, and some feedback antibodies cannot match the self strings. As the result, some detectors were built up. They made some matches against the incoming antigens, and the detection rates dropped to around 99%. The best result was achieved when $R = 35$, the detection rate was 99.21%, with error rate 0.79%.

Combining both attack strings and normal strings together, the best results were under the setting of either $R = 33$ or $R = 34$, when the activation matching threshold was set at 5, Fig 2.

These results are surprisingly good for the antigen feedback mechanism. They prove its effectiveness. With the feedback, the system can quickly establish useful detectors and then use them to discover similar patterns from further incoming antigens.

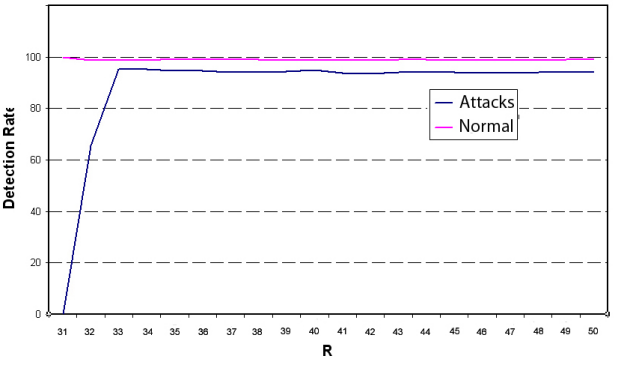


Fig. 2. The detection rates for R from 31 to 50 with the activation matching threshold 5

The matching method, r-continuous bits match, was inspired by real immune systems [7, 11]. We believe that it can accurately measure the similarity of two strings, because it measures the degree of *resembleness* instead of distance, such as Euclidean distance or Hamming distance. For example, the Euclidean distance distances for the two pairs (i) "10101" and "00100" and (ii) "10101" and "11111" are the same ($\sqrt{2}$), but the two strings in the first pair resembles each better than the second pair, as the 3 continuous bits in the middle are the same (010).

Given the good detection rates we achieved under the antigen feedback mechanism, we still have some concerns on using segmented string format to represent antibodies and antigens for r-continuous bits match method. Suppose that the length of antibodies and antigens is l, and a string has *n* segments from s_1 to s_n. If $R = l$, we have full matching, where a detector can only detect the antigens which are exactly the same as itself. If $\frac{l}{2} < R < l$, r-continuous bits match just ignores the leading or the ending, or both, segments, e.g., s_1, s_2, s_{n-1}, and s_n etc., depending on the value of R and the lengths of the leading and the ending segments. Therefore, the method makes the segments in the middle weight more than these at the both ends. If a match happens, these in the middle have to have exact match. In other words, a match involves all the segments in the middle and some from either end of the string. Therefore, a match is achieved by completely and also consistently ignored some segments at either ends. This is the reason responsible for our string format as described in Section 3.

This effect is not desirable. However, the problem is not the matching method, but the way we present the data. We are seeking a better way to represent antibodies and antigens by homogenizing the information carried by each segment.

6 Conclusion and Future Work

This paper presents our experiment on negative selection in intrusion detection by using KDD CUP 1999 dataset. In order to solve the problem of efficiently generating

effective detectors, we propose an antigen feedback mechanism. Under the mechanism, we can achieve 95.21% detection rate for attack strings, with false negative rate 4.79%, and 99.21% detection rate for normal strings, with false positive rate 0.79%.

The surprisingly good results by such a simple mechanism were achieved on the KDD CUP 1999 dataset. More experiment on other datasets, and perhaps the datasets from different domains, is needed to verify if the results are just coincident or persistent. We are arranging more experiment. Given the good results achieved by the antigen feedback mechanism and the fact that randomly generated antibodies hardly match the incoming antigens, we have a bold conjecture that negative selection may not need randomly generated antibodies at all but just rely on the feedback antigens. If the conjecture can be proven to be true in general cases, we avoid the scaling problem of negative selection.

This paper also introduces Arisytis, the test bed we used for our experiment. It is available to the other researchers in the field to avoid the waste of the effort to write similar program. Arisytis can also be used as an educational tool.

Finally, we believe that negative selection with the antigen feedback mechanism has the ability to quickly discover unknown patterns. The ability can be applied to other domains for pattern detection than just self/non-self discrimination, for example, spam email recognition, faulty parts detection, and finance fraud discovery etc. Another future task of ours is to find out a way to homogenously blend the information carried by the characters of the antibody and antigen strings and then study the differences of r-continuous bits match on the new string format. Last but not least, we will further expand Arisytis with other up to date AIS algorithms.

References

1. Forrest, S., Hofmeyr, S.A., et al.: A sense of self for Unix processes. In: IEEE Symposium on Security and Privacy, Oakland, CA, USA (1996)
2. Timmis, J.: Artificial immune systems - today and tomorrow. Natural Computing: an international journal 6(1), 1–18 (2007)
3. Dasgupta, D.: Advances in artificial immune systems. IEEE Computational Intelligence Magazine 1(4), 40–49 (2006)
4. Garrett, S.M.: How Do We Evaluate Artificial Immune Systems? Evolutionary Computation 13(2), 145–177 (2005)
5. Dasgupta, D., Ji, Z., Gonzalez, F.: Artificial immune system (AIS) research in the last five years. In: The 2003 Congress on Evolutionary Computation (CEC 2003). IEEE Press, Los Alamitos (2003)
6. Hofmeyr, S.A., Forrest, S.: Immunity by Design: An Artificial Immune System. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999), Orlando, Florida. Morgan Kaufmann, USA (1999)
7. Hofmeyr, S.A., Forrest, S.: Architecture for an Artificial Immune System. Evolutionary Computation 8(4), 443–473 (2000)
8. Hart, E., Timmis, J.: Application Areas of AIS: The Past, The Present and The Future. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627. Springer, Heidelberg (2005)

9. Forrest, S., Perelson, A.S., et al.: Self-Nonself Discrimination in a Computer. In: Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA. IEEE Computer Society Press, Los Alamitos (1994)
10. Hofmeyr, S.A., Forrest, S., Somayaji, A.: Intrusion Detection Using Sequences of System Calls. Journal of Computer Security 6, 151–180 (1998)
11. Hofmeyr, S.: An Immunology Model of Distributed Detection and Its Application to Computer Security. Department of Computer Science, University of New Mexico, USA (1999)
12. Castro, L.N.D., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
13. Balthrop, J., Forrest, S., Glickman, M.R.: Revisiting LISYS: Parameters and normal behavior. In: Proceedings of the Congress on Evolutionary Computing (CEC-2002) (2002)
14. Gabrielli, N., Rigodanzo, M.: An Artificial Immune System for Network Intrusion. Detection on a Web Server: First Results. In: Proceedings of the 2nd Italian Workshop on Evolutionary Computation (GSICE 2006) (2006)
15. Gonzalez, F.A., Dasgupta, D.: Anomaly Detection Using Real-Valued Negative Selection. Genetic Programming and Evolvable Machines 4(4), 383–403 (2003)
16. Ji, Z., Dasgupta, D.: Revisiting Negative Selection Algorithms. Evolutionary Computation 15(2), 223–251 (2007)
17. Kim, J., Bentley, P.: An evaluation of negative selection in an artificial immune system for network intrusion detection. In: Proceedings of GECCO 2001 (2001)
18. ACM. KDD CUP 1999 data. [cited 12 January 2007], http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html
19. DARPA. DARPA Intrusion Detection Evaluation Data Sets. 1999 [cited 2006 15 October 2006], http://www.ll.mit.edu/IST/ideval/data/data_index.html
20. Stolfo, S.J., Fan, W., et al.: Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project. In: Proceedings of 2000 DARPA Information Survivability Conference and Exposition (2000)
21. Ma, W., Tran, D., Sharma, D.: A Study on the Feature Selection of Network Traffic for Intrusion Detection Purpose. In: The Proceedings of IEEE International Conference on Intelligence and Security Informatics (ISI 2008) (to be published, 2008)

A Neuro-Immune Algorithm to Solve the Capacitated Vehicle Routing Problem

Thiago A.S. Masutti and Leandro N. de Castro

Laboratory of Intelligent Systems (LSIn)
Catholic University of Santos (UNISANTOS)
144, Dr. Carvalho de Mendonça St., Vila Mathias
Santos/SP, Brazil, 11070-906
{tmasutti,lnunes}@lsin.unisantos.br

Abstract. Some features of a large number of combinatorial optimization problems prevent the use of exact solution methods, thus requiring the application of heuristic techniques to find good solutions, not always the optimal ones, in a feasible amount of time. This paper describes a heuristic approach, which is a hybrid between artificial neural networks and artificial immune systems, to solve the capacitated vehicle routing problem. This algorithm is based on a competitive model, which does not use a cost or evaluation function to determine the quality of the solution proposed. Despite this apparent drawback, the set of tests conducted with the proposed approach indicates a good performance of the algorithm when compared with similar works from the literature and the known best solutions available.

Keywords: Artificial Neural Networks, Artificial Immune Systems, Self-Organizing Maps, Combinatorial Optimization, Capacitated Vehicle Routing Problem.

1 Introduction

Natural Computing has played an important role in combinatorial optimization by providing good heuristic approaches [1]. One of these methods is based on Artificial Neural Networks that, for combinatorial optimization tasks, can be divided into two types: 1) networks based on the optimization of an error surface, such as the Hopfield Networks [2]; and 2) self-organized networks, such as the Elastic Net [3] and Self-Organizing Feature Maps [4]. The second type of networks is the one with the best results in the literature [5].

The work of Hopfield & Tank [2] was pioneer in using artificial neural networks to solve combinatorial problems, in particular the traveling salesman problem (TSP), using a Hopfield Network. Concerning self-organizing networks, a pioneer work was presented by Durbin & Willshaw [3] with the Elastic Net, and then Fort [6] and Angeniol et al. [7] with the use of self-organizing maps (SOM), all applied to solving the Traveling Salesman Problem (TSP). Although many works from the literature

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 210–219, 2008.

discuss the application of self-organizing networks to solve the TSP [8], [9], little attention has been paid to more complex problems, such as the Capacitated Vehicle Routing Problem (CVRP).

The CVRP is described as follows. A fixed number k of identical vehicles, each with capacity Q, is available at the depot. There are M customers (plus the depot), each with a demand q_i (i = 1, 2, 3, ..., M), to be served. The CVRP consists of determining k routes with minimum costs, so that: 1) each route starts and ends at the depot; 2) each client is visited once by a single vehicle; and 3) the demand of the customers served by a vehicle does not exceed the capacity of the vehicle. The CVRP is an NP-complete problem [10], preventing the use of exact algorithms for certain instances of the problem, and thus requiring the use of heuristic approaches.

This paper proposes one such heuristic based on artificial immune systems and self-organizing maps to solve the CVRP, named RABNET-CVRP. This algorithm is an extension of [11], which solves the TSP. Concepts from SOM and artificial immune systems are used to generate several sub-networks to compose a feasible solution to the problem. Several tests are conducted with instances commonly used in the literature and a comparison is performed with other algorithms based on self-organizing networks. The RABNET-CVRP is an algorithm with unsupervised learning, that is, it does not use a cost or evaluation function to determine the quality of the solution. Although this may look like a limitation, the results achieved show a good performance of the proposed method.

The remainder of this paper is organized as follows. Section 2 presents an overview of a few works relating self-organizing networks to solve the CVRP; in Section 3 the proposed algorithm, RABNET-CVRP, is detailed; the computational results are presented in Section 4; and the paper is concluded in Section 5 with a brief discussion about the current and further works.

2 Related Works

Few works based on self-organizing networks were proposed to solve the CVRP. This section makes a brief review of these works, emphasizing those that will be used for performance comparisons with the proposed algorithm.

In the work of Vakhutinsky & Golden [12], the authors propose an extended version of the Elastic Net [3] to solve the CVRP. In this algorithm, several sub-networks, each one representing a vehicle, are expanded in the direction of the cities. The number of neurons is predetermined and their weights are iteratively updated according to two rules: 1) one that moves a neuron in the direction of the closest city, avoiding the violation of the vehicle's capacity; and 2) a rule that moves a neuron in the direction of its nearest neighbors, aiming at minimizing the length (tour) of the network. The performance of the algorithm is evaluated with a set of 5 instances ranging in size from 22 to 51 cities. The best solutions found do not violate the capacity constraints and they are compared with the best known solution for each

instance. The authors argue that the results are not as good as those of the best techniques from operations research, but even so they are satisfactory.

In [13] the authors propose an algorithm based on self-organizing maps to solve the CVRP. In the proposed algorithm, several sub-networks stretch from the depot into the direction of the customers, where each sub-network represents the route of a vehicle. For the algorithm to be able to solve a CVRP instance there is a *bias* term for each network, which reflects the demand covered by the represented vehicle and it is used along the competition process avoiding the construction of routes that exceed the vehicle's capacity. In the proposed approach, the number of neurons is fixed as three times the number of customers to be served. To assess the performance of the proposed algorithm, a set of 10 instances ranging in size from 22 to 200 cities was used and the results compared with the best known solution for each instance and to the results presented in [12]. According to the results, the proposed algorithm outperformed the proposal of [12].

In the work of Gomes & Von Zuben [14], the authors propose a hybrid algorithm based on self-organizing maps and fuzzy systems to solve the CVRP. The sub-networks' architecture is modified along the training phase according to a mechanism of insertion and pruning neurons based on the algorithm of Angeniol et al. [7]. A fuzzy module that acts upon the competition rule provides the construction of low cost routes handling the other constraints of the problems, such as the vehicle's capacity. To assess the performance of the proposed algorithm, it was used a set of 7 instances ranging in size from 22 to 101 cities. The results are compared with the best known solution for each instance and with an algorithm without the fuzzy module based on [13]. According to the results presented, the proposed algorithm outperformed the one without the fuzzy module.

3 RABNET-CVRP

The RABNET-CVRP (*real-valued antibody network to solve the capacitated vehicle routing problem*) is a heuristic approach that combines concepts from artificial immune systems [15] and self-organizing maps [4] to solve the CVRP. This algorithm is an extended version of [11] and [16], which solve the TSP, and of [17], which solves the MTSP. The main characteristics of the RABNET-CVRP are: 1) *feedforward* neural network with no hidden layer; 2) competitive network with unsupervised learning; 3) constructive architecture with growing and pruning phases; and 4) pre-defined circular neighborhood.

The goal of RABNET-CVRP is, throughout the learning phase, to position one network cell near enough each city of the CVRP instance to be solved. In RABNET-CVRP, there are several sub-networks, each one representing the route of a vehicle. Thus, at the end of the learning phase, the pre-defined neighborhood of each sub-network will denote the sequence of cities to be covered by the vehicles. The main steps of RABNET-CVRP are described in the following.

3.1 Sub-networks Initialization

As each sub-network represents one vehicle, k sub-networks are initialized. In each network, there is a number of antibodies (in artificial neural networks, the analogue for an antibody is a neuron) equal to round(M/k) where one antibody (the antibody related to the depot) has an attribute vector (weight vector in the neural network literature) equal to the coordinate vector of the depot and the other antibodies are randomly distributed on the Euclidean plane. In order to facilitate the implementation, the antibody related to the depot, at each sub-network, has index one.

3.2 Presentation of Antigens (Cities)

During the immune system evolution, an organism can meet a certain antigen several times [15]. As the problem to be solved by the self-organizing network is the CVRP, each city corresponds to one antigen (input pattern in artificial neural networks) and they are iteratively presented to the antibody network, simulating the meeting between the organism and an antigen. Prior to each epoch, the order of the cities is randomized so as to avoid that this order influences the network adaptation.

3.3 Competition

This step consists of determining the winner antibody (city) to the presented antigen according to the following equation:

$$I, J = \arg\min \left\{ \begin{array}{ll} \left\| \mathbf{ag} - \mathbf{Ab}_i^j \right\| \cdot \left[1 + 0{,}5 \cdot \left({qt_j}/{Q} \right) \right] & 0.8 < qt_j \le Q \\ \left\| \mathbf{ag} - \mathbf{Ab}_i^j \right\| \cdot \left(1 + {qt_j^2}/{Q} \right) & qt_j > Q \\ \left\| \mathbf{ag} - \mathbf{Ab}_i^j \right\| & \text{otherwise} \end{array} \right\}, \quad (1)$$

where **ag** is the coordinate vector of the presented antigen, $\mathbf{Ab}_i^j$ is the attribute vector of antibody i belonging to sub-network j, qt is the sum of the presented city's demand with the demand covered by vehicle j, and I is the index of the winner antibody belonging to the sub-network J. One can observe that the competition rule depends on two terms: 1) the (Euclidean) distance between the presented antigen and the attribute vector of an antibody; and 2) the current demand covered by a given vehicle. The main objective of this second term is to penalize vehicles that cover a high demand ($0.8 < qt_j \le Q$) and to inhibit vehicles that cover a demand higher than their capacities ($qt_j > Q$).

An antibody can be a winner for zero, one or more antigens. A vector $\boldsymbol{\rho}_j$ stores, for each sub-network, the number of antigens related to each antibody. This information is reinitialized at each epoch and is used in other steps, such as the Growing Phase.

If the antigen presented represents the depot, there is no competition. The antibodies of index one, for each sub-network, will be used in the next steps. Thus, one can say that for the depot, there are k winners.

3.4 Cooperation

The cooperation step is based on self-organizing maps [4]. At this step, the stimulus to the winner antibody is propagated to its neighbors, but with a smaller intensity. In RABNET-CVRP, there is no connection between the sub-networks, so the neighborhood of the winner antibody is restricted to the sub-network to which it belongs.

This stimulus' intensity h_{iI} for antibody i belonging to sub-network J is computed by the following equation:

$$h_{iI} = \begin{cases} \exp\left[-d_{iI}^{\,2} / 2 \cdot \sigma(t)^2\right] & i \neq 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $\sigma(t)$ is the parameter that controls the influence of the neighborhood, t is the current epoch and d_{iI} is the neighborhood degree between the winner antibody I and antibody i given by the following equation, which defines a circular neighborhood:

$$d_{iI} = \min\left(|i - I|;\quad N - |i - I|\right),\ i \in J \tag{3}$$

where N is the number of antibodies in sub-network J.

The second term of Equation (2) ensures that the attribute vector of the antibody related to the depot will not change, being always equal to the coordinates of the depot.

The influence of the neighborhood has to be large at the beginning of the learning phase and be reduced throughout the epochs [4]. In RABNET-CVRP this is done by updating $\sigma(t)$ according to the following equation:

$$\sigma(t) = \sigma(0) \cdot \exp(-t/\tau_1), \tag{4}$$

where $\sigma(0)$ is its initial value and $\tau_1 = 100$.

3.5 Adaptation

To present an effective response to pathogenic agents, the immune system counts with a learning process that involves the increasing of the antigen-antibody affinity at each meeting of the organism with a determined antigen. In RABNET-CVRP, the adaptation phase performs this increase in antigen-antibody affinity by moving the winner antibody and its neighbors in the direction of the antigen presented to the network according to the following equation:

$$\mathbf{Ab}_i^J(t+1) = \begin{cases} \mathbf{Ab}_i^J(t) + \alpha(t) \cdot h_{iI} \cdot \left[\mathbf{ag} - \mathbf{Ab}_i^J(t)\right] & h_{iI} > \kappa \\ \mathbf{Ab}_i^J(t) & \text{otherwise} \end{cases}, \tag{5}$$

where κ is a threshold for h_{il} and $\alpha(t)$ is the learning rate, redefined at each epoch by the following equation:

$$\alpha(t) = \alpha(0) \cdot \exp(-t/\tau_2), \tag{6}$$

where $\alpha(0)$ is its initial value and $\tau_2 = 300$.

The threshold κ in Equation (5) defines a minimum value for h_{il} to antibody i to be updated. This restriction limits the effective neighborhood of an antibody, allowing only significant updates. According to preliminary tests, this restriction does not affect significantly the quality of the solutions, but saves a reasonable amount of processing time.

3.6 Sub-networks Growing

In this step, the most stimulated antibody from each sub-network along one epoch is selected for cloning. The most stimulated antibody is the one recognizing the higher concentration of antigens, defined by the following equation:

$$C_j = \text{argmax}(\boldsymbol{\rho}_j), \tag{7}$$

where C_j is the index of the most stimulated antibody from sub-network j, and $\boldsymbol{\rho}_j$ is the concentration vector of this sub-network.

With a selected antibody for cloning, among all antigens related to it, that with the highest Euclidean distance is selected. If this distance is greater than a predefined threshold ε, then this antibody is cloned. Otherwise, no change occurs in this sub-network architecture. The attribute vector of the newly created antibody is the same as the one from its parent antibody, and its neighborhood is of degree one in relation to its parent.

3.7 Convergence Criteria

Two criteria define the convergence of the algorithm to a solution of the selected problem: 1) the antibodies of all sub-networks must be related to at most one antigen; and 2) each antigen u ($u = 1, 2, \ldots, M$) must have one antibody related to it at a minimum distance λ, and this antibody must be the current winner for antigen u. If these conditions are satisfied, the learning process is finished. It is not necessary to test the second criterion with the depot, since there is always one antibody from each sub-network related to it and those antibodies have attribute vectors equal to the depot's coordinates.

3.8 Pruning

At the end of the learning process, every antibody not related to any antigen is removed from the sub-networks. It makes the number of antibodies equal to $M + k$.

Fig. 1 illustrates the flowchart for RABNET-CVRP.

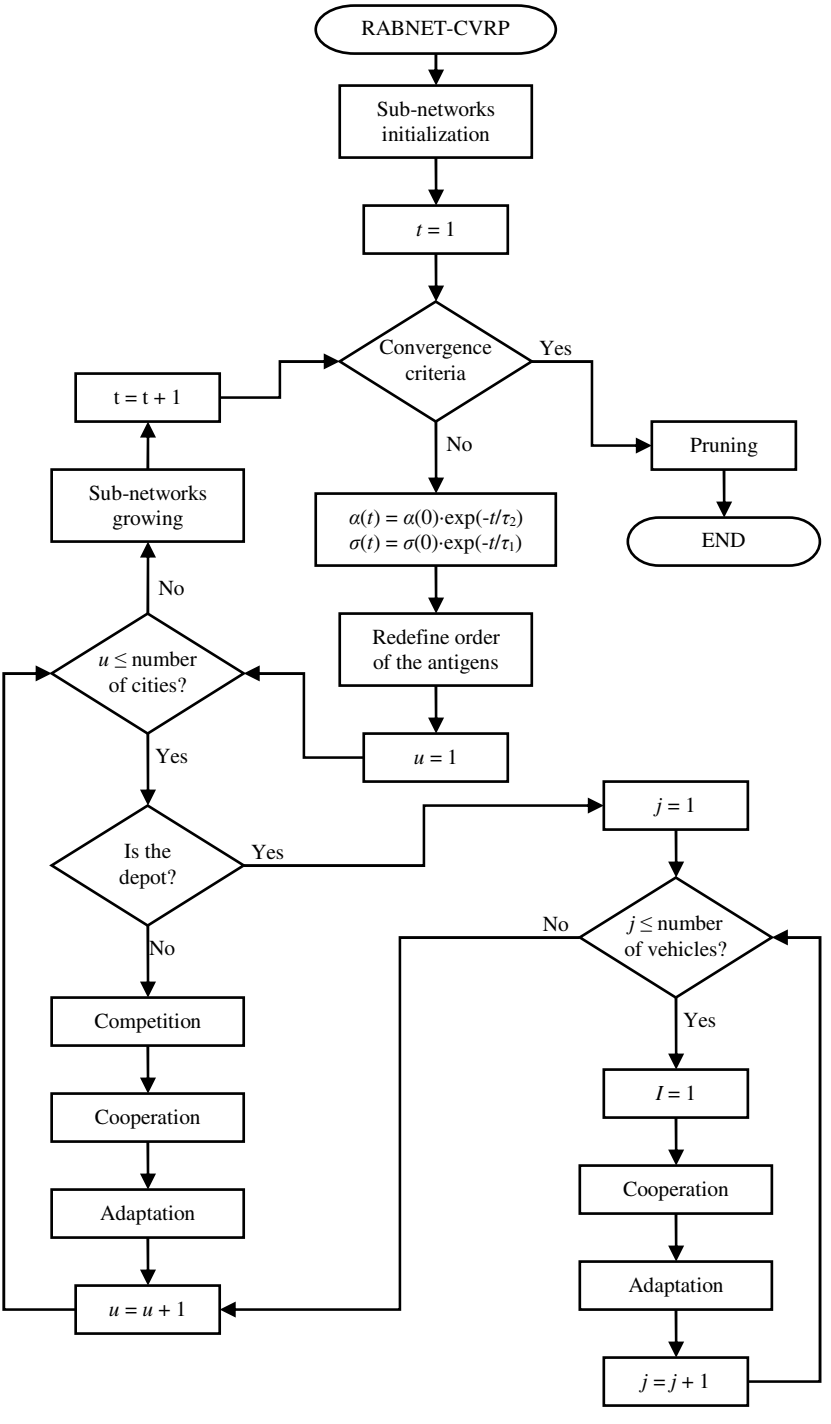


Fig. 1. Flowchart for the RABNET-CVRP

4 Computational Results

To assess the performance of the algorithm, several experiments were conducted with instances commonly used in literature. These instances were taken from [18] and are available at http://www.branchandcut.org/VRP/data/. The algorithm was coded in MATLAB and run in a PIV 3,0 GHz with 1GB RAM.

The proposed algorithm has a set of parameters to be tuned before being applied to a CVRP instance. The parameters $\alpha(0)$ e $\sigma(0)$ are those that most influence the behavior of the sub-networks and, consequently, the quality of the solutions found. Appropriate values for these parameters vary according to the instance to be solved and can be obtained with preliminary experiments. In cases this is not possible, the following default values (obtained empirically) are suggested for all parameters: $\alpha(0) = 0.75$, $\sigma(0) = 40$, $\varepsilon = md*0.2$, $\lambda = md*0.4$, $\kappa = 0.01$, $\tau_1 = 100$ e $\tau_2 = 300$, where *md* represents the minimum distance among all cities.

The tests were conducted with a set of 9 instances ranging in size from 22 to 101 cities. For each instance, the proposed algorithm was run 50 times and the results obtained are compared with the best known solutions and with results presented in other works dealing with self-organizing networks to solve the CVRP [12], [13], [14].

Table 1 presents the computational results for RABNET-CVRP regarding the quality of solutions, demonstrated by the best solution found, the average solution and the effectiveness in finding feasible solutions; and regarding to the computational effort, demonstrated by the average number of epochs and antibodies for the convergence of the algorithm and the running time. Table 2 presents a comparison between the best solutions found by RABNET-CVRP and other three similar works.

Table 1. Computational results for RABNET-CVRP regarding the computational effort and the quality of solutions found. BKS is the best known solution; PCV is the percentage of solutions that violates the capacity constraint; Epochs and NA are the average number of epochs and antibodies, respectively; Time is the average running time in seconds; Best is the cost of the best solution found that does not violate the capacity constraint; Mean is the average cost of solutions that do not violate the capacity constraint. All results shown were taken from 50 runs for each instance.

Instance	BKS	PCV	Epochs	NA	Time	Best	Mean
E-n22-k4	375	12	204.84	335.50	2.76	375	388.3
E-n30-k3	534	48	162.76	175.28	2.38	543	559.7
E-n33-k4	835	20	267.22	439.82	5.67	876	899.5
E-n51-k5	521	48	157.18	276.68	5.33	578	632.1
E-n76-k7	682	08	251.80	738.40	17.98	692	708.4
E-n101-k8	815	14	267.66	1029.70	29.90	839	865.3
Att-n48-k4	40002	02	256.94	475.72	7.93	40212	40991
F-n45-k4	724	40	267.66	496.22	8.10	735	767.2
F-n72-k4	237	42	187.68	416.14	8.55	254	294.5

Table 2. Comparison of the quality of the solutions found by RABNET-CVRP with the solutions presented in other three similar works: Vakhutinsky & Golden [12], named VG; Torki et al. [13], named TSE; and Gomes & Von Zuben [14], named GVZ. BKS is the best known solution and PDB is the percent deviation of the best solution found to the best known solution.

Instance	BKS	PDB			
		RABNET-CVRP	VG	TSE	GVZ
E-n22-k4	375	0.00	75.73	4.00	2.27
E-n30-k3	534	1.69	60.11	4.31	2.79
E-n33-k4	835	4.91	7.07	6.47	4.64
E-n51-k5	521	10.94	7.49	3.07	0.91
E-n76-k7	682	1.47	-	-	-
E-n101-k8	815	2.94	-	5.89	7.55
Att-n48-k4	40002	0.52	-	-	-
F-n45-k4	724	1.52	-	-	-
F-n72-k4	237	7.17	-	-	-

5 Conclusion and Future Investigation

This paper presented a heuristic approach hybridizing artificial neural networks with artificial immune systems to solve the CVRP. The CVRP is one of the most studied combinatorial optimization problems in the literature, justifiable by its wide applicability in real world problems and by its difficulty in solving, since most often the capacity constraint conflicts with the route minimization objective.

The proposed algorithm, named RABNET-CVRP, solves a CVRP instance by positioning one antibody from any sub-network (each one representing a vehicle) near enough each city of the instance, and the antibodies' neighborhood defines the sequence of cities to be visited by each vehicle. The main characteristics of RABNET-CVRP are its unsupervised competitive model and constructive architecture. Its performance was evaluated with a set of standard data and its results were directly compared with other three similar works and the best known solutions from the literature.

According to the results obtained, RABNET-CVRP showed to be capable of finding good quality solutions with a short computational effort. For the set of instances used in this paper, the proposed algorithm presented an average percent deviation of 3.46% from the best known solutions. A direct comparison of the best solutions obtained by RABNET-CVRP with solutions presented in other three similar works demonstrated that the proposed algorithm obtained competitive solutions, outperforming them, with regards to the quality of the solution found, on three out of five instances. However, it is important to stress that RABNET-CVRP was not capable of finding feasible solutions, regarding the capacity constraint, in all runs. For the set of instances used in this paper, the percentage of feasible solutions was greater than 50%.

Further investigations might include computational tests with a larger number of instances, a model which presents a higher percentage of feasible and good quality

solutions and the joint use of the proposed approach with an improvement heuristic, such as simulated annealing and *k*-opt.

Acknowledgement

The authors thank Fapesp and CNPq for the financial support.

References

1. de Castro, L.N.: Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications. Chapman & Hall/CRC (2006)
2. Hopfield, J.J., Tank, T.W.: "Neural" Computation of Decisions in Optimization Problems. Biological Cybernetics 52(3), 141–152 (1985)
3. Durbin, R., Willshaw, D.: An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Method. Nature 326, 689–691 (1987)
4. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Berlin (2001)
5. Smith, K.A.: Neural Networks for Combinatorial Optimization: A Review of More than a Decade of Research. INFORMS Journal on Computing 11(1), 15–34 (1999)
6. Fort, J.C.: Solving a Combinatorial Problem via Self-Organizing Process: An Application of the Kohonen Algorithm to the Traveling Salesman Problem. Biological Cybernetics 59(1), 33–40 (1988)
7. Angeniol, B., Croix Vaubois, G., Le Texier, J.-Y.: Self-Organizing Feature Maps and the Traveling Salesman Problem. Neural Networks 1, 289–293 (1988)
8. Somhom, S., Modares, A., Enkawa, T.: A Self-Organising Model for the Travelling Salesman Problem. Journal of the Operational Research Society 48(9), 919–928 (1997)
9. Cochrane, E.M., Beasley, J.E.: The Co-Adaptive Neural Network Approach to the Euclidean Travelling Salesman Problem. Neural Networks 16(10), 1499–1525 (2003)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
11. Pasti, R., de Castro, L.N.: A Neuro-Immune Network for Solving the Traveling Salesman Problem. In: International Conference on Neural Networks, pp. 3760–3766 (2006)
12. Vakhutinsky, A.I., Golden, B.L.: Solving Vehicle Routing Problems Using Elastic Nets. In: IEEE International Conference on Neural Networks, pp. 4535–4540 (1994)
13. Torki, A., Somhom, S., Enkawa, T.: Competitive Neural Network Algorithm for Solving Vehicle Routing Problem. Computer & Industrial Engineering 33(3-4), 473–476 (1997)
14. Gomes, L.C.T., Von Zuben, F.J.: Vehicle Routing Based on Self-Organization with and without Fuzzy Inference. In: IEEE International Conference on Fuzzy Systems, pp. 1310–1315 (2002)
15. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, London (2002)
16. Masutti, T.A.S., de Castro, L.N.: A Constructive Self-Organizing Network Applied to a Discrete Optimization Problem. In: Seventh International Conference on Intelligent Systems Design and Applications, pp. 52–57 (2007)
17. Masutti, T.A.S., de Castro, L.N.: Uma Abordagem Neuro-Imune para a Solução do Problema de Múltiplos Caixeiros Viajantes. In: VIII Brazilian Conference on Neural Networks (CD-ROM) (2007)
18. Christofides, N., Eilon, S.: An Algorithm for the Vehicle Dispatching Problem. Operational Research Quarterly 20(3), 309–318 (1969)

Improving Artificial Immune System Performance: Inductive Bias and Alternative Mutations

Pupong Pongcharoen[1,*], Warattapop Chainate[1], and Sutatip Pongcharoen[2]

[1] Department of Industrial Engineering, Faculty of Engineering,
[2] Department of Medicine, Faculty of Medicine, Naresuan University,
Pitsanulok, Thailand 65000
pupongp@nu.ac.th, pupongp@yahoo.com

Abstract. Research works related to the Artificial Immune System (AIS) and their applications have been extensively reported during the last decade. In this work, we proposed an inductive bias heuristic called neighbourhood improvement within the classical AIS for improving its performance. We also demonstrated alternative mutation mechanisms for cloning the elite antibodies. Computational experiments using the proposed heuristic and mechanisms to find the near optimal solutions of travelling salesman problems were conducted. The results obtained from the modified AIS were compared with those obtained from other metaheuristics. It was found that the performance of the modified AIS adopting the proposed heuristic and mechanisms outperformed the conventional AIS and other metaheuristics.

Keywords: Artificial Immune System, Genetic Algorithms, Particle Swarm Optimisation, Simulated Annealing, Tabu Search, Travelling Salesman.

1 Introduction

Optimisation algorithms can be categorised as being either conventional or approximation optimisation algorithms [1, 2]. Conventional optimisation algorithms are usually based upon mathematical models such as Integer Linear Programming [3], Branch and Bound [4] or Dynamic Programming [5]. These approaches were relatively well developed and attributed to the military services early in World War II. Based on the full enumerative search within these approaches, the optimal solutions are always guaranteed. However, the application of these methods might need exponential computational time in the worst case. This becomes an impractical approach especially for solving a very large size problem. Alternative approaches that can guide the search process to find near optimal solutions in acceptable computational time are therefore more practical and desirable.

Approximation optimisation algorithms so called metaheuristics have therefore received more attention in the last few decades. Metaheuristics iteratively conduct stochastic search process inspired by natural intelligence. They can be categorised into three groups [6]: physically-based inspiration such as Simulated Annealing [7];

* Corresponding author.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 220–231, 2008.

socially-based inspiration for instance Tabu Search [8]; and biologically-based inspiration e.g. Neural Network [9], Genetic Algorithms [10], Shuffled Frog Leaping [11], Particle Swarm Optimisation [12], Ant Colony Optimisation [13] and Artificial Immune System [14]. These alternative approaches have been widely used to solve large-scale combinatorial optimisation problems [14-17].

Research works related to the Artificial Immune System (AIS) and their applications have been extensively reported during the last decade [18-20] and also discussed in the ICARIS conferences. However, the immune inspired algorithms have been limited by the lack of theoretical advances, the adoption of a naive immune inspired approach and the limited application of AIS to challenging problems [20]. Freitas and Timmis [21] have pointed out that there is a lack of appreciations for possible inductive bias within the AIS algorithms and positional bias within the choice of representation and affinity measures. The algorithms might be tailored by embedding heuristic information related to a specific problem as an inductive bias procedure for improving its performance.

Mutation mechanisms play an important role in the cloning process conducted within the Artificial Immune System algorithms. The cloning process performs an exploration principle for searching candidate solutions in the solution space. Two mutation mechanisms called inverse and pairwise interchange mutations are generally adopted in the clonal section AIS algorithms [22, 23]. Other mutation mechanisms adapted within the Genetic Algorithms (GA) have also been systematically investigated and reported in literature [24, 25]. Pongcharoen et al. [24] have investigated eleven mutation mechanisms used in the GA for solving travelling salesman problem. The analysis on their experimental results suggested that the best mutation mechanism was the Shift Operation Mutation [26] followed by the Inversion Mutation [10].

The objectives of this paper were to: i) statistically investigate the appropriate setting of the percentage of antibody elimination; ii) demonstrate the use of alternative mutation mechanisms (Shift Operation and Inversion Mutations) for improving the performance of the classical Artificial Immune System; and iii) propose an inductive bias process called Neighbourhood Improvement (NI) heuristic acting as the heuristic information related to the travelling salesman problem.

This paper is organised as follows. Section 2 describes the proposed AIS algorithm for travelling salesman problem. Section 3 presents the design and analysis of computational experiments for identifying the appropriate setting of the percentage of antibody elimination and investigating the performance of the modified AIS with alternative mutation mechanisms and heuristic against other metaheuristics in terms of the quality of the results obtained and the execution time required. The conclusions are drawn in section 4 followed by appendices and references.

2 Artificial Immune System

The biological immune system is a defending system of living organisms. In humans, the immune system is highly developed with its ability to distinguish self from non-self. The immune system functions by detecting and recognising the non-self or foreign molecules that enter the body (e.g. infectious micro-organisms or transplanted tissues). These functions confer by phagocytes as well as antigen-presenting cells

such as B cells and dendritic cells. These cells then specifically present antigens derived from the foreign bodies that they had internalised to T cells via specific T cell receptors on the T cell membrane. The specifically stimulated T cell then responses by proliferating giving rise to antigen-specific T cell clones. Each clone has T cells with the same specificity to the stimulating antigen.

The B cells specifically bind antigens using membrane receptors called B cell receptors. The antigen-bound B cells are then activated to proliferate and also to become 'antibody-producing' plasma cells. The proliferating B cell clones and the antibodies secreted from plasma cells all have the same specificity to the stimulating antigen. The proliferation rate of a B cell is directly proportional to its recognising degree of the antigen. The B cell learns by raising the population size and affinity (the degree of the cell recognition with the antigen) [27].

Artificial Immune System (AIS) is one of the biology-inspired method, which is a branch of computational intelligence [28]. The artificial immune system is based on two main principles [22]: clonal selection and affinity maturation principles. In the first principle, each antibody (candidate solution) has an affinity (fitness) value determined by the affinity (objective) function. The latter principle consists of two main processes: mutation and receptor editing. Mutation mechanisms such as inverse mutation and/or pairwise interchange mutation can be used to generate a clone from an antibody [23]. The number of clones is determined by its affinity value and the size of antibody population. After cloning, sorting and deleting the repetition, the receptor editing process is conducted by eliminating antibodies from the population based on the desired percentage of antibody elimination (*%B*). The whole process is repeated until the termination criterion is satisfied. A pseudo code for the clonal selection AIS algorithm is provided in the appendices. However, there have been other mutation mechanisms previously proposed within Genetic Algorithm [25].

In this work, we proposed alternative mutation mechanisms called Shift Operation Mutation [26] and Inversion Mutation [10], both of which have been statistically proven to be the effective mutation mechanisms within the GA [24]. These mechanisms were therefore proposed to replace the inverse and pairwise interchange mutations as a modified clonal selection AIS (MAIS). In addition, we proposed a heuristic called Neighbourhood Improvement (NI) within the modified AIS by taking the best affinity antibody to perform an inductive bias process during the evolutional iterations. The pseudo code for the NI heuristic is shown in the appendices. The heuristic uses the distance between cities to find the best solution from neighbour solutions within iteration. The best antibody is then cloned in the next iteration. The integration of the AIS algorithm, mutation schemes and NI heuristic is provided in the pseudo code shown in the appendices.

3 Experimental Design and Analysis

Travelling salesman problem (TSP) is one of the classical combinatorial optimisation problems. The problem is to minimise the cost of the tour travelled by a salesman who want to visit every cities in his territory only once and return to the starting city. The cost of the tour is basically determined by the length of the itinerary travelled. In this work, three symmetric travelling salesman problem (TSP) instances provided in

the TSP library [29] were adopted in the numerical experiments. The first problem instance was relatively small and named as "pr76", in which the number indicates the number of 76 cities to be travelled. The remaining problems named "bier127" and "a280" were relatively medium and large, respectively.

Two step sequential experiments were carried out. In the first experiment (A), was aimed to investigate the appropriate setting of AIS parameter, the percentage of eliminating antibodies *(%B)* through a 76 nodes of the TSP instance. The finding of appropriate setting of *%B* was sequentially applied in the next experiment, which was aimed to benchmark the performance of the AIS algorithms with other metaheuristics. In the latter experiment (Experiment B), the total number of search determining by the amount of candidate solutions multiplied by the number of iterations was fixed to 5,000 generated solutions. The higher values of these parameters mean that there is more chance of getting good solutions but this requires longer computational time. The computational experiment was repeated 30 times using different random seed numbers. The simulation program including the proposed algorithms was coded in modular style using Microsoft Visual Basic 6.0. All computational runs were conducted on a PC with Intel Core2Duo 2.66 GHz 2.0 GB RAM.

Experiment A

It is understood that no algorithm can outperform others for all problem instances and domains due to the distinct nature of the problems and its complexity. In fact, the performance of the algorithm depends on its parameters' setting. For example, one of the AIS parameters called the percentage of eliminating antibody (*%B*) has been specified at 10% by Agarwal et al. [30] for solving project scheduling problem but 30% has been used to solve job shop scheduling problem [22]. This experiment was aimed to investigate the appropriate setting of the percentage (*%B*) for solving the travelling salesman problem instance (pr76). The values of *%B* were ranged from 0-30%. In this experiment, the number of antibodies (*P*) and the number of iterations (I_{max}) were set to 10 and 500, respectively. The experimental results obtained with 30 replications for each value of *%B* were analysed by using the general linear model form of analysis of variance (ANOVA) shown in Table 1. It can be seen that the percentage (*%B*) was statistically significant with 95% confidence level (having *p* values less than 0.05).

Table 1. ANOVA table of the computational results in the experiment A

Source	Degree of Freedom	Sum of Square	Mean Square	F	*p*
%B	3	2356174422	785391474	3.41	0.020
Error	116	26753968144	230637656		
Total	119	29110142566			

The main effect plots shown in Figure 1 suggested that the average distance of the tours produced by the AIS having *%B* at 10% was shortest compared to those with other settings. Therefore, this finding on *%B* was sequentially adopted in the next experiment.

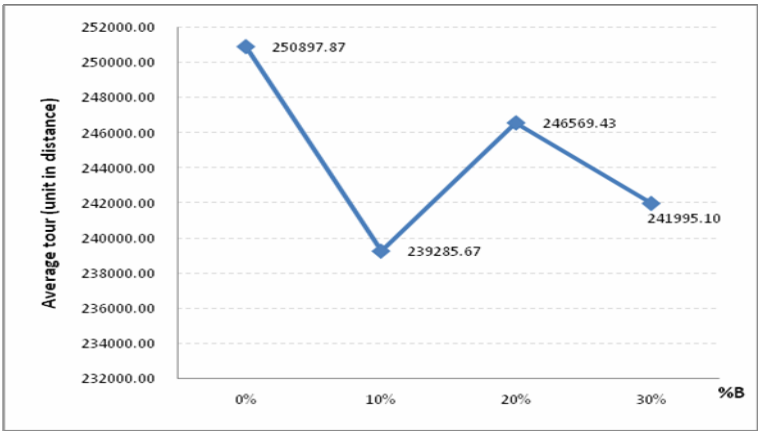


Fig. 1. Average tour obtained from AIS by varying %B factors

Experiment B

This sequential experiment was aimed to benchmark the performance of conventional Artificial Immune System (AIS) with the modified AIS (MAIS) using alternative mutation mechanisms mentioned in the previous section, MAIS that embedded the proposed inductive bias heuristic called Neighbourhood Improvement (MAIS+NI), Genetic Algorithm (GA), Tabu Search (TS), Simulated Annealing (SA_{Geo}) with Geometric cooling scheme [7], SA with Lundy-Mees cooling schedule (SA_{LM}) [31] and Particle Swarm Optimisation (PSO) in terms of quality of solutions obtained and computational time required. The pseudo codes of these algorithms are provided in the appendices.

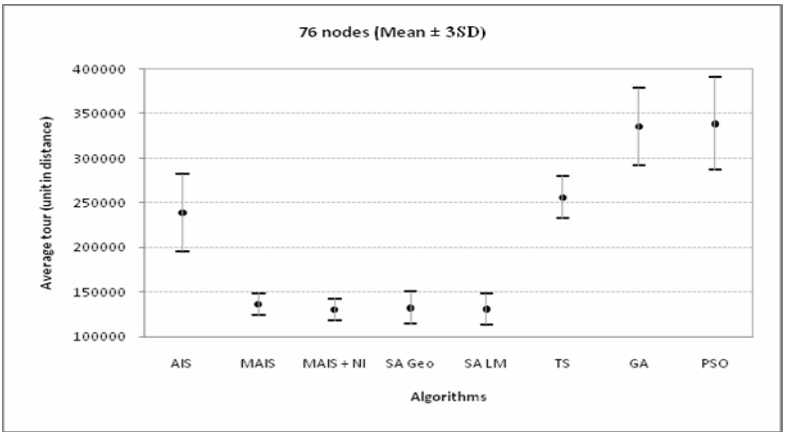
From the finding in experiment A, the setting of percentage of eliminating antibodies *(%B)* for all AIS algorithms was 10% of the population size of antibody. The mutation operators used in conventional AIS has been Inverse and Pairwise interchange mutations, but the experimental results from the previous research [24] have suggested that the mutation operators suitable for solving TSP are Shift operation (SOM) and Inversion (IM) mutations. These mutation operators, SOM and IM, were therefore applied in MAIS and MAIS+NI. It should be noted that the operation of Inversion mutation is not the same as the Inverse mutation used in AIS.

The setting of other well-known algorithms was set based on the pervious suggestions. The setting of both SA parameters was based on the suggestion from Glass and Potts [32], in which the value of initial and final temperatures of SA_{LM} factors were 500 and 10, respectively and 700 and 30 for SA_{Geo}. For TS parameters, taboo list size was determined by the number of nodes divided by 3 and the candidate list per iteration was set to 5 [33]. The setting of GA parameters was based on the suggestion from our previous research [24, 25, 33, 34], in which the value of the combination of population size and the number of generations (*P*/*G*), probabilities of crossover (P_c) and mutation (P_m), crossover (COP) and mutation operators (MOP) were 100/50, 0.9, 0.5, edge recombination (ERX) and shift operation (SOM), respectively. The setting of PSO factors including the value of the combination of particle size and the number of iterations (N/I_{max}), c_1 and c_2 were 100/50, 0.1 and 0.9, respectively.

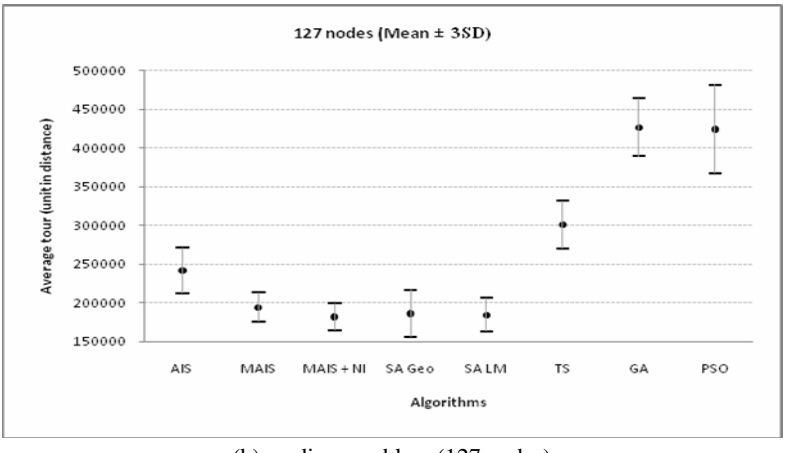
The computational runs using each algorithm were repeated 30 times with different random seed numbers for each TSP instant dataset. The high number of replications can reduce the effect of stochastic transition rules conducted within the algorithms. Due to the stochastic transition rules conducted within those algorithms, the measures of performance were considered on the mean and the standard deviation (SD) of the solutions obtained. The experimental results of 720 runs were analysed in terms of the average, standard deviation (SD.), maximum and minimum distance of the tours obtained from each method including its execution time (second) as shown in Table 2. It can be seen that the average distance of the tours obtained from the modified AIS (MAIS) was considerably better than those using the conventional AIS, TS, GA and PSO for all problem sizes. But the performance of the MAIS was outperformed by the SA_{Geo} and SA_{LM}. Considering the modified AIS with Neighbourhood Improvement (MAIS+NI), it can be seen that the average tour distance obtained from MAIS+NI marginally outperformed those results using other metaheuristics for small and medium problems. For the large problem, the average tour distance produced by the MAIS+NI was shorter than those using SA_{LM} and SA_{Geo} by 23% and 54%, respectively. The best so far tours for small, medium and large size problems were found by SA_{LM}, SA_{Geo} and MAIS+NI, respectively. The average execution time taken by PSO was quickest followed by SA, AIS and GA. Figure 2 shows the mean and the error bar (±3SD) of the best tours obtained from all approaches for each problem sizes.

Table 2. The experimental results of all algorithms for experiment B

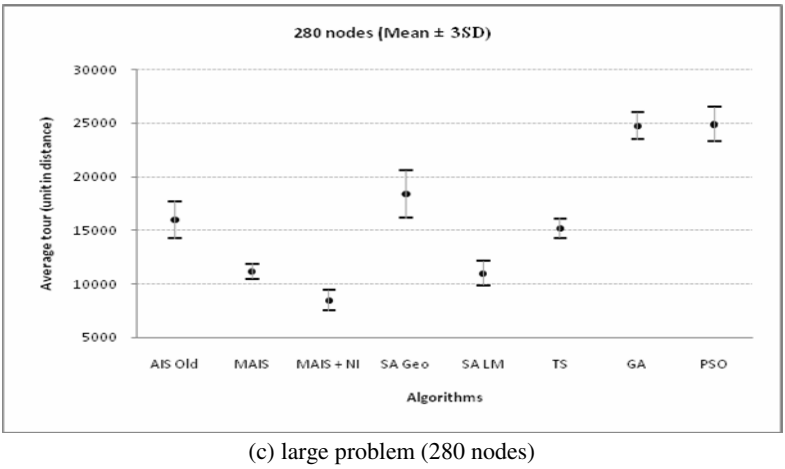
Problem size	Algorithms	Quality of solutions (tours) obtained				Computational Time (s)
		Average	SD.	Min	Max	
Small (76 Nodes)	AIS	239285.67	14423.04	216093	277455	10.07
	MAIS	136821.20	4193.55	127438	143004	9.97
	MAIS+NI	**130886.17**	**4187.74**	**123977**	**140303**	10.97
	SA_{Geo}	132879.50	6106.33	124327	147140	2.37
	SA_{LM}	131502.83	5807.55	121860	144478	2.40
	TS	256817.63	7900.13	238711	271945	3.63
	GA	335746.97	14503.49	311912	363064	18.70
	PSO	339141.60	17299.69	308331	384315	1.47
Medium (127 Nodes)	AIS	245261.37	9759.87	223260	261075	14.63
	MAIS	195079.10	6269.90	180596	207933	14.57
	MAIS+NI	**182738.73**	**5759.61**	**170559**	**195591**	16.40
	SA_{Geo}	186804.63	10070.2	168399	207136	3.87
	SA_{LM}	185121.47	7209.46	173411	205168	3.93
	TS	301507.93	10290.19	283377	325694	6.30
	GA	427210.33	12364.31	402190	446432	52.00
	PSO	424979.03	18896.63	387955	463034	2.53
Large (280 Nodes)	AIS	16021.40	576.92	14638	17021	29.57
	MAIS	11222.50	**238.55**	10740	11742	30
	MAIS+NI	**8487.17**	319.32	**7669**	**9126**	36.07
	SA_{Geo}	18427.33	736.47	16089	19497	8.90
	SA_{LM}	11030.17	381.64	10184	11643	8.87
	TS	154234.03	302.06	14616	15963	14.70
	GA	24798.20	422.04	23972	25497	311.57
	PSO	24940.47	530.84	23966	25888	6.20



(a) small problem (76 nodes)



(b) medium problem (127 nodes)



(c) large problem (280 nodes)

Fig. 2. Average tours produced by each method for all instant datasets

4 Conclusions

Research works related to the Artificial Immune System (AIS) and their applications have been extensively reported during the last decade. However, it has been reported that the immune inspired algorithms have been limited by the lack of theoretical advances and appreciations for possible inductive bias within the algorithms. In this present work, we proposed an inductive bias heuristic called Neighbourhood Improvement within the clonal selection AIS for improving its performance. We also demonstrated alternative mutation mechanisms for cloning the elite antibodies. Computational experiments using the proposed heuristic and mechanisms to find the near optimal solutions using benchmarking travelling salesman problems were conducted. The results obtained from the modified AIS were compared with those obtained from other metaheuristics including Simulated Annealing, Tabu Search, Genetic Algorithm and Particle Swarm Optimisation. It was found that the performance of the modified AIS adopting both proposed heuristic and mechanisms considerably outperformed the conventional AIS and other metaheuristics especially for the large size problem.

Acknowledgements. This work was supported by the Thailand Research Fund and the Naresuan University Research Fund.

References

1. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing surveys 35, 268–308 (2003)
2. Nagar, A., Haddock, J., Heragu, S.: Multiple and bicriteria scheduling: A literature survey. European Journal of Operational Research 81, 88–104 (1995)
3. Chen, K., Ji, P.: A mixed integer programming model for advanced planning and scheduling (APS). European Journal of Operational Research 181, 515–522 (2007)
4. Brucker, P., Knust, S., Schoo, A., Thiele, O.: A branch and bound algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research 107, 272–288 (1998)
5. Choi, J., Realff, M.J., Lee, J.H.: Dynamic programming in a heuristically confined state space: a stochastic resource-constrained project scheduling application. Computers & Chemical Engineering 28, 1039–1058 (2004)
6. Engin, O., Doyen, A.: Artificial immune systems and applications in industrial problems. G. U. Journal of Science. 17, 71–84 (2004)
7. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimisation by simulated annealing. Science 220, 671–679 (1983)
8. Glover, F.: Tabu search - part I. ORSA Journal on Computing 1, 190–206 (1986)
9. Haykin, S.: Neural networks: A comprehensive foundation, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)
10. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley, Massachusetts (1989)
11. Eusuff, M., Lansey, K., Pasha, F.: Shuffled frog-leaping algorithm: a memetic metaheuristic for discrete optimization. Engineering Optimization 38, 129–154 (2006)
12. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)

13. Dorigo, M., Stutzle, T.: Ant Colony Optimization. Bradford Book, Massachusetts (2004)
14. Hart, E.A., Timmis, J.: Application areas of AIS: The past, the present and the future. Applied Soft Computing 8, 191–201 (2008)
15. Aytug, H., Knouja, M., Vergara, F.E.: Use of genetic algorithms to solve production and operations management problems: a review. International Journal of Production Research 41, 3955–4009 (2003)
16. Chaudhry, S.S., Luo, W.: Application of genetic algorithms in production and operations management: a review. International Journal of Production Research 43, 4083–4101 (2005)
17. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical Computer Science 344, 243–278 (2005)
18. Dasgupta, D.: Artificial Immune Systems and Their Applications. Springer, Heidelberg (1998)
19. De Castro, L.: Artificial Immune Systems: Theory and Applications. In: Brazilian Symposium on Neural Networks, Rio de Janeiro, Brazil (2000)
20. Timmis, J.: Artificial Immune Systems - today and tomorrow. Natural Computing 6, 1–18 (2007)
21. Freitas, A., Timmis, J.: Revisiting the Foundations of Artificial Immune Systems: A Problem-Oriented Perspective. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 229–241. Springer, Heidelberg (2003)
22. Chandrasekaran, M., Asokan, P., Kumanan, S., Balamurugan, T., Nickolas, S.: Solving job shop scheduling problems using artificial immune system. International Journal of Advanced Manufacturing Technology 31, 580–593 (2006)
23. Engin, O., Doyen, A.: A new approach to solve hybrid flow shop scheduling problems by artificial immune system. Future Generation Computer Systems 20, 1083–1095 (2004)
24. Pongcharoen, P., Chainate, W., Thapatsuwan, P.: Exploration of genetic parameters and operators through travelling salesman problem. Science Asia 33, 215–222 (2007)
25. Pongcharoen, P., Stewardson, D.J., Hicks, C., Braiden, P.M.: Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry. Journal of Applied Statistic 28, 441–455 (2001)
26. Murata, T., Ishibuchi, H.: Performance evaluation of genetic algorithms for flowshop scheduling problems. In: Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 812–817 (1994)
27. Murphy, K., Travers, P., Walport, M.: Janeway's Immunobiology. Garland Science (2007)
28. Dasgupta, D.: Advances in artificial immune systems. IEEE computational intelligence magazine, 40-49 (November 2006)
29. TSPLIB. Travelling salesman problem library, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
30. Agarwal, R., Tiwari, M.K., Mukherjee, S.K.: Artificial immune system based approach for solving resource constraint project scheduling problem. International Journal of Advanced Manufacturing Technology 34, 584–593 (2007)
31. Lundy, M., Mees, A.: Convergence of an annealing algorithm. Mathematical Programming 34, 111–124 (1986)
32. Glass, C.A., Potts, C.N.: A comparison of local search methods for flow shop scheduling. Annals of Operations Research 63, 489–509 (1996)
33. Azimi, Z.N.: Hybrid heuristics for Examination Timetabling problem. Applied Mathematics and Computation 163, 705–733 (2005)
34. Pongcharoen, P., Promtet, W.: Exploring and determining genetic algorithms parameters through experimental design and analysis. In: Proceedings of the 33rd international conference on computers and industrial engineering, Jeju, Korea (2004)

Appendices

Appendix A. Pseudo Code for AIS Procedure

Initialise the value of AIS parameters e.g. antibody size (P), iterations (I_{max}) and percentage of antibody elimination ($\%B$).
Generate a population of P antibodies
For each antibody ($i \in P$), calculate affinity (i)
Set current iteration (I) = 1
Do
    For each antibody (i)
        Calculate the number of clones (N_C) and clone antibody (i)
        For each clone, apply inverse mutation to create a new antibody
            Calculate the affinity of the new antibody
            If affinity (new antibody) is better than the clone,
            Then clone = new antibody,
            Else
                Perform pairwise interchange mutation to create a new antibody
                Calculate the affinity of the new antibody
                If affinity (new antibody) is better than the clone,
                Then clone = new antibody,
            End
        End
        antibody (i) = clone
    End
    Eliminate the worst antibodies from the population based on $\%B$
    Create new antibodies to replace the eliminated antibodies
    $I = I + 1$
While $I \leq I_{max}$

Appendix B. Pseudo Code for a SA Procedure

Initialise the value of SA parameters including starting temperature (t_0), cooling rate (α) and final temperature (t_{min}).
Generate a candidate solution (s)
Calculate the fitness (energy) value of the current solution, $E(s)$
Set current temperature (t) = t_0
Do
    Do
        Find a neighbour solution (s^*) of the current solution (s)
        If $E(s^*)$ is better than $E(s)$ or (random [0,1) $\leq \exp((E(s)-E(s))/t)$), $s = s^*$
    While stopping criteria is satisfied
    $t = \alpha t$
While $t \leq t_{min}$

Appendix C. Pseudo Code for Neighbourhood Improvement (NI) Procedure

Begin
    Select a candidate solution (tour)
    Identify the worse pair of cities in the tour
    Relocate both cities in the tour by checking the new tour's distance
    If the new tour is better than the previous tour, then tour = new tour
End

Appendix D. Pseudo Code for MAIS Procedure

Initialise the value of AIS parameters e.g. antibody size (P), iterations (I_{max}) and percentage of antibody elimination ($\%B$).
Generate a population of P antibodies
For each antibody ($i \in P$), calculate affinity (i)
Set current iteration (I) = 1
Do
    For each antibody (i)
        Calculate the number of clones (N_C) and clone antibody (i)
        For each clone, apply Inversion Mutation (IM) to create a new antibody
            Calculate the affinity of the new antibody
            If affinity (new antibody) is better than the clone,
            Then clone = new antibody,
            Else
                Perform Shift Operation Mutation (SOM) to create a new antibody
                Calculate the affinity of the new antibody
                If affinity (new antibody) is better than the clone,
                Then clone = new antibody,
            End
        End
        antibody (i) = clone
    End
    Eliminate the worst antibodies from the population based on $\%B$
    Create new antibodies to replace the eliminated antibodies
    In case of MAIS+NI, apply Neighbourhood Improvement (NI) for the best antibody
    $I = I + 1$
While $I \leq I_{max}$

Appendix E. Pseudo Code for a TS Procedure

Initialise the value of TS parameters e.g. length of tabu list (L), iteration (I_{max}).
Generate a candidate solution (s)
Calculate the fitness value of the current solution, $f(s)$
Set taboo list is empty
Set current iteration (I) = 1
Do
    Find a set of neighbour solutions of the current solution
    Select the best neighbour solution (s^*) from the set of neighbour solutions
    If $f(s^*)$ is better than $f(s)$, $s = s^*$
    Update taboo list
    $I = I + 1$
While $I \leq I_{max}$

Appendix F. Pseudo Code for a GA Procedure

Initialise the value of GA parameters e.g. population size (P), number of generations (G) and probabilities of crossover (P_c) and mutation (P_m).
Generate a population of P chromosomes
For each chromosome ($i \in P$), calculate fitness (i)
Set current generation (g) = 1
Do
    Based on P_c, randomly select two parent chromosomes for crossover operation
    Based on P_m, randomly select a parent chromosome for mutation operation
    Calculate the fitness of the offspring

If offspring is better than the parent, replace the parent
$g = g + 1$
While $g \leq G$

Appendix G. Pseudo Code for a PSO Procedure

Initialise the value of PSO parameters including swarm size (N), number of iterations (I_{max}), inertia weight (ω), self (c_1) and social (c_2) learning rates.
Generate a swarm of N particles
For each particle ($i \in N$), calculate fitness (i)
Set current iteration (I) = 1
Do
For each particle, update the best fitness of particle (i) as *pBest*
Update the best fitness of all particles as *gBest*
For each particle
Calculate particle velocity based on the *pBest* and *gBest*
Update particle position based on the new velocity
End
$I = I + 1$
While $I \leq \mathrm{I}_{\max}$

Flexible Immune Network Recognition System for Mining Heterogeneous Data

Mazidah Puteh[1], Abdul Razak Hamdan[2], Khairuddin Omar, and Azuraliza Abu Bakar

[1] Universiti Teknologi MARA, Terengganu, Malaysia
[2] Universiti Kebangsaan Malaysia, Bangi, Malaysia

Abstract. Artificial Immune System (AIS) is an emerging technique for the classification task and proved to be a reliable technique. In previous studies, many classifiers including AIS classifiers require the data to be in numerical or categorical data types prior to processing. The transformation of data into any other specific types from their original form can degrade the originality of the data and consume more space and pre processing time. This paper introduces AIS model using immune network for classifying heterogeneous data in its original types. The model is able to process the data with the types as represented in the database and it solves some bias problems highlighted in the AIS review papers. To ensure the consistent conditions and fair comparison, the selected existing algorithms use the same set of data as used in the proposed model. Experimental results show that this network-based model produces a better accuracy rate than the existing population-based immune algorithm and than the standard classifiers on most of the data from University of California, Irvive (UCI) Machine Learning Repository (MLR) and University of California, Riverside (UCR) Time Series Data (TSR).

Keywords: artificial immune system (AIS), classification, immune network, heterogeneous, accuracy, significant difference.

1 Introduction

Data mining serves a task of extracting knowledge from real world data sets. It helps in decision making and forecasting on the currently available knowledge or information. A classification is one of important tasks in data mining to help people make a better decision in the future based on the available knowledge. With the available classification algorithm, people can repeatedly make a forecast on the accumulated knowledge in new situations.

The proposed model in this paper focuses on the classification task of data mining. More precisely, this work proposes an algorithm that will classify heterogeneous data (data with the combination of various types such as continuous, discrete and nominal) without the need for the transformation of the data into any specific type. Transforming the data into different type from its original form somehow will degrade the accuracy of the data [26] and this consumes more space

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 232–241, 2008.

and data pre processing time. Some limitations in AIS models as highlighted by [11] [13] [20] are the data representation and suitable affinity measurements that cause inductive bias problem. Furthermore, previous studies in AIS classifiers did not test enough real world dataset with low accuracy rate when using standard algorithm. Note also that this study concerns the computational AIS and does not suppose to offer any insight into theoretical immunology.

The technique used in this paper is to induce a set of classification rules based on immune network of artificial immune system (AIS), a relatively new computational intelligence paradigm [10]. The advantage of using AIS technique is that it uses a distance function to find similarity or affinity measurement between two instances. This allows the technique to incorporate the hybrid affinity measurement so called heterogeneous value difference metric (HVDM) distance function that is mainly for evaluating dataset with heterogeneous types with special methods to handle outliers and bias value.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of natural immune system, artificial immune system and affinity measurements. Section 3 generally explains the proposed algorithm Flexible Immune Network Recognition System (FINERS). Section 4 reports the experiments and result discussion. Finally, section 5 presents the conclusion and future work for the proposal.

2 Immune System

A biological immune system has two broad response systems. One is innate immunity, which is general and not normally modeled by AIS systems. The other one is an adaptive immunity that is based on two kinds of antibody cells in the body: T-cells, so named because they originate in the thymus gland and B-cells originate in bone marrow. When a pathogen invades the body, special cells called antigens are available. An individual T-cell or B-cell responds to the antigens by cloning and mutating to match the antigen. The affinity of that T-cell or B-cell from the antigen [15] is stronger if the the match is closer.

B-cells that do not match any antigens and do not stimulate with the neighbour antibodies eventually die. When a body has successfully defended against a pathogen, a comparatively small number of memory cells remain in the body for very long time. These memory cells recognize antigens similar to those that originally cause the immune response, so that the body's response to a future and very similar invader is much faster and powerful than to a never-before-seen invader.

2.1 Artificial Immune System (AIS)

An Artificial Immune System is a bio-inspired computational model that uses ideas and concepts from the natural immune system, mainly the interaction between antigen and B-cells (stimulation and suppression), interaction between

antibody and antibody from the immune network theory [19], cloning and mutation process [9]. It offers strong and robust information processing capabilities for solving complex problems. Applications of AIS include supervised and unsupervised machine learning, pattern recognition, intrusion detection and security [8].

Among the early models on supervised machine learning are Immunos81 [4] and AIRS [22] [23]. However, the former model uses a significantly different and complex approach. The later model is the first straightforward immune-inspired supervised learning algorithm and has subsequently gone through a period of study and refinements [12] [24] [25]. However, many of these studied classifiers concentrate on the population-based AIS algorithm and ignore the important network feature [19] of the immune system. They also require numerical representation of data, use Euclidean affinity measurement and mostly are tested only on numerical dataset.

As suggested in [11] [13] [20] [22], methods of using other types of data need to be explored to allow for greater applicability of this learning paradigm. [12] have explored variety of distance metric for affinity measurements with population-based AIS algorithm but a more comprehensive experiment on many problems with heterogeneous types is required in order to proof a high quality classification technique for heterogeneous data types using hybrid affinity measurement. To overcome the limitation and improve the classification accuracy, there is a need for developing the AIS classifier with the network feature and be able to accept heterogeneous data without the need for the data transformation. In order to accept heterogeneous types of data, all processes involving these data must consider appropriate and suitable affinity measurement, data structure and mutation method.

2.2 Affinity Measurements

One of the important components in AIS framework is affinity measurement. The affinity measurement in AIS uses the distance metric function such as Euclidean function, Value Difference Metric (VDM) and Heterogeneous Value Difference Metric (HVDM). There are many learning systems depend on good distance function to measure similarities and be successful such as the nearest neighbor techniques [6] [7] [14], and memory-based reasoning methods [18]. Such algorithms have had much success on a wide variety of applications (real-world classification tasks).

The Euclidean distance function works well with numerical attribute [26] but do not handle nominal attributes appropriately. The VDM [18] was introduced to define an appropriate distance function for nominal attributes. It works well in many nominal domains, but it does not handle continuous attributes directly. Instead, they rely upon process of discretization which can degrade generalization accuracy [21].

Many real-world applications have both nominal and numeric attribute as shown in the UCI MLR [17]. The HVDM was introduced in [26] that can take

heterogeneous data where it uses normalized VDM for nominal data and normalized difference for linear data. HVDM has shown a good potential to be the distance metric for heterogeneous data without the need for any discretization or transformation of data into any specific type. HVDM has become the choice as the hybrid affinity measurement for the AIS algorithm in this paper.

Heterogeneous Value Difference Metric (HVDM). As mentioned in the previous section, the Euclidean distance function is inappropriate for nominal attributes, and VDM is inappropriate for continuous attribute, so neither is sufficient on its own for use on a heterogeneous application, i.e. one with both nominal and continuous attributes. So, HVDM is used as shown below:

$$HVDM\,(x,y) = \sqrt{\sum_{a=1}^{m} {d_a}^2\,(x_a, y_a)} \tag{1}$$

where m is the number of attributes. The function $d_a(x,y)$ returns a distance between the two values x and y for attribute a and it is defined as:

$$d_a(x,y) = \left\{ \begin{array}{ll} 1, & \text{if } x \text{ or } y \text{ is unknown; otherwise} \\ vdm_a\,(x,y)\,, & \text{if } a \text{ is nominal} \\ diff_a\,(x,y)\,, & \text{if } a \text{ is linear} \end{array} \right\} \tag{2}$$

where vdm and $diff$ are defined as follows:

$$vdm_a\,(x,y) = \sqrt{\sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^2} \tag{3}$$

where

- $N_{a,x}$ and $N_{a,y}$ are the numbers of instances in the training set that have value x and y for attribute a;)
- $N_{a,x,c}$ and $N_{a,y,c}$ are the numbers of instances in the training set that have the value x and y for attribute a and output class c;
- C is the number of output classes in the problem domain;

and

$$diff_a\,(x,y) = \frac{|x-y|}{4\sigma_a} \tag{4}$$

where x and y are input linear values and σ is the standard deviation for attribute a.

Distances are often normalized by dividing the distance for each variable by the range of that attribute, so that the distance for each input variable is in

the range 0..1 and this is employed by algorithm in [12]. However, dividing by the range allows outliers (extreme values) to have a profound effect on the contribution of an attribute. A more robust alternative in the presence of outliers is to divide the values by the standard deviation to reduce the effect of extreme values on the typical cases [26].

The situation for HVDM is more complicated because the nominal and numeric distance values come from different types of measurements: numeric distances are computed from the differences between two linear values, normalized by standard deviation, while nominal attributes are computed from a sum of C differences of probability values (where C is the number of output classes). It is therefore necessary to find a way to scale these two different kinds of measurements into approximately the same range to give each variable a similar influence on the overall distance measurement. The detail experiment and explanation for evaluating HVDM can be found in [26].

3 Proposed Algorithm Flexible Immune Network Recognition System (FINERS)

In the real world situation, there are many data set which comprise of both numerical and nominal data types. This paper investigates the use of hybrid affinity measurement in immune network algorithm for applying heterogeneous datasets that are composed of nominal, discrete or continuous data types or the combination of them without the need for the transformation of the data into any specific type. The algorithm in the proposed model considers an appropriate data structure to suit the complexity of recognizing heterogeneous data in its original types.

The algorithm FINERS works as follows:

```
1st STAGE:
-Calculate Affinity Threshold (AT) by calculating average
   affinity (distance) between all pairs among antigens
-MemoryCell (MC) initialization, usually starts with null
    For each antigen do
    2nd STAGE:
    -Search for mcmatch from MC, if unavailable, antigen as
       mcmatch
    -Clone and mutate mcmatch
    -Generate first generation antibodies (AB)
    -Create a network among antibodies with affinity greater
        than network affinity threshold (NAT)
    3rd STAGE:
    -Clone and mutate antibody from AB randomly until average
       stimulation is greater than stimulation threshold.
    -Generate the final AB
```

```
 -Create a network among antibodies with affinity greater
    than network affinity threshold (NAT)
4th STAGE:
-Search for mccandidate (most stimulated) from AB
-Compare mccandidate to mcmatch, if mccandidate is more
   stimulated, it is added to MC. If affinity between
   mccandidate and mcmatch is less than
   affinity threshold scalar times affinity threshold
   then mccandidate replaces mcmatch inside MC
-Create a network among antibodies with affinity greater
     than network affinity threshold (NAT)
```

4 Experiment and Discussion

Experiment is carried out on eleven datasets from UCI MLR [17]. The datasets are carefully selected to represent different data types such as real-valued IRIS and Ionosphere (ION), Nominal Zoo and Tic-tac-toe (TTT), Discrete Wisconsin Breast Cancer (WBC) and Lung Cancer (LC) and Heterogeneous Australian Credit (CRX), German Credit (GC), Hepatitis (HP), Cleveland Heart Disease (HD) and Ljubljana Breast Cancer (BC) [28]. The experiment is also carried out on five time series datasets from UCR TSD [16]. The time series data are Beef, ECG200 (ECG), Lighting7 (LIGHT7), OSULeaf (OSU) and Synthetic Control (SC). The dataset is distributed into 10 fold cross validation with 90 percent data for training and 10 percent data for testing with no overlapping. The data are tested in their original types as provided in the databases.

For a consistent condition and comparison on FINERS with other immune and standard classification algorithms, WEKA toolbox [27] is used to run the same sets of 10-fold CV data used in FINERS. The selected classifiers from WEKA toolbox are the immune algorithms with detail explanation in [27] as the representatives of earlier population-based AIS models: AIRS1 [1] [23], AIRS2 [1] [24], AIRS2 Parallel (AIRS2P) [1] [25], CLONALG [2] [9], Immunos1 [3], Immunos2 [3], and Immunos99 [3]. The standard algorithms are chosen as a representative from different categories include Nave Bayes (NB) from bayesian category, Multilayerperceptron (MLP) and Learning Vector Quantization (LVQ) from neural net, IBK from instance-based category, J48/C45 from tree and NNGE from nearest-neighbour category. The descriptions of these classifiers can be studied from [27].

The average accuracy is calculated from the 10 sets for each dataset and the significant difference is analyzed using paired T-Test in standard statistical package. The importance of the significant difference test is to show whether the difference can be taken into consideration or the difference is too small to be considered and can just be assumed as comparable. The difference is significant only if the significant value is less than 0.05 with 95 percent confidence [5].

The accuracy rates from the experiment comparing FINERS and previous immune classifiers on heterogeneous and non-heterogeneous data with the significant value of differences are shown in Table 1.

Table 1. Comparison of the accuracy rates between FINERS and the other immune algorithms on heterogeneous and non-heterogeneous data with their significant value of differences in 2nd column labeled with SIG. SIG value < 0.05 shows that the differences are significant.

		Real-valued		Nominal		Discrete		Heterogeneous				
	SIG	**IRIS**	**ION**	**ZOO**	**TTT**	**WBC**	**LC**	**CRX**	**GC**	**HD**	**HP**	**BC**
FINERS		97	88	89	91	98	50	87	75	73	89	93
AIRS1	**0.036**	96	86	98	90	97	36	80	67	68	82	83
AIRS2	**0.002**	94	84	89	84	96	34	83	71	68	82	84
AIRS2P	**0.013**	94	82	98	83	96	44	81	71	67	80	85
CLONALG	**0.010**	92	71	94	64	94	54	63	70	68	71	75
Immunos1	**0.027**	98	71	96	67	85	48	85	68	71	86	80
Immunos2	**0.001**	97	62	55	64	67	37	74	71	69	76	80
Immunos99	**0.002**	96	70	78	66	82	45	82	70	71	86	82

The accuracy rates from the experiment comparing FINERS and other standard classifiers on heterogeneous and non-heterogeneous data with the significant value of differences are shown in Table 2.

Table 2. Comparison of the accuracy rates between FINERS and the standard classifiers on heterogeneous and non-heterogeneous data with their significant value of differences in 2nd column labeled with SIG. SIG value > 0.05 shows that the differences are not significant.

		Real-valued		Nominal		Discrete		Heterogeneous				
	SIG	**IRIS**	**ION**	**ZOO**	**TTT**	**WBC**	**LC**	**CRX**	**GC**	**HD**	**HP**	**BC**
FINERS		97	88	89	91	98	50	87	75	73	89	93
NB	**0.159**	97	82	98	68	96	51	77	76	74	84	87
MLP	**0.311**	96	93	96	97	95	41	86	75	64	83	82
LVQ	**0.007**	95	83	91	76	95	48	64	67	64	76	69
IBK	**0.221**	96	88	98	99	95	49	83	70	62	79	83
J48/C4.5	**0.125**	97	89	95	85	94	47	86	76	73	78	76
NNGE	**0.053**	97	89	96	83	96	47	83	74	64	78	81

The accuracy rates from the experiment comparing FINERS and previous immune classifiers on time series data with the significant value of differences are shown in Table 3.

Accuracy rate comparing FINERS and other standard classifiers on time series data with the significant value of differences is shown in Table 4.

On heterogeneous and non-heterogeneous data sets, the results in Table 1 and Table 2 show that FINERS produces better accuracy rates than the rates produced by all previous immune algorithms and LVQ from the standard classifiers. The SIG value between FINERS and these algorithms falls below 0.05. For other standard classifiers, FINERS produces comparable accuracy rates because the SIG value for comparing FINERS and these standard classifiers is more than 0.05.

Table 3. Comparison of the accuracy rates between FINERS and the other immune algorithms on time series data with their significant value of differences in 2nd column labeled with SIG. SIG value < 0.05 shows that the differences are significant.

	SIG	BEEF	ECG	LIGHT7	OSU	SC
FINERS		62	95	67	65	93
AIRS1	**0.009**	47	75	36	24	79
AIRS2	**0.035**	40	73	58	22	87
AIRS2P	**0.003**	40	71	51	33	81
CLONALG	**0.012**	47	79	51	40	51
Immunos1	**0.001**	42	73	49	40	59
Immunos2	**0.011**	30	64	30	26	89
Immunos99	**0.000**	32	71	46	38	57

Table 4. Comparison of the accuracy rates between FINERS and the standard classifiers on time series data with their significant value of differences in 2nd column labeled with SIG. SIG value > 0.05 shows that the differences are not significant.

	SIG	BEEF	ECG	LIGHT7	OSU	SC
FINERS		62	95	67	65	93
NB	**0.079**	37	74	61	32	98
MLP	**0.002**	47	81	46	38	66
LVQ	**0.087**	50	88	56	42	97
IBK	**0.065**	62	91	63	64	91
J48/C4.5	**0.025**	58	78	61	43	81
NNGE	**0.035**	57	80	59	47	59

On time series data sets, the results in Table 3 show that FINERS produces better accuracy rates than the rates produced by all previous immune algorithms with the SIG value < 0.05. The results in Table 4 show that FINERS produce better accuracy rates than MLP, C4.5 and NNGE with the SIG value between FINERS and these algorithms falls below 0.05. For other standard classifiers, FINERS produces comparable accuracy rates because the SIG value is more than 0.05.

5 Conclusion

This paper has proposed a new AIS network-based classifier that is called Flexible Immune NEtwork Recognition System (FINERS) that uses hybrid affinity measurement for heterogeneous data type without the need for the discretization or transformation of the data into any specific type. The experimental results show that the immune network-based model produces better accuracy rates in most of the heterogeneous, non-heterogeneous and time series datasets compared to population-based immune classifiers. The results also show that FINERS is better than some standard classifiers and comparable to some. It shows that the

study solves some limitation shown in [11] [13] [20] [22]. The results suggest that it is significant to process the data in its original types to avoid degradation of data accuracy from transformation, discretization and normalization of original data prior to processing. Avoiding some of these data preprocessing task will decrease the preprocessing time and space.

For future investigation, other AIS algorithm can employ hybrid affinity measurements for other tasks such as optimization and clustering. FINERS could also be further refined to make it a dynamic algorithm which can process dynamic data. With the result, we hope to derive a more stable and flexible AIS classifier.

References

1. Brownlee, J.: Artificial Immune Recognition System (AIRS) A review and Analysis. CISCP, Faculty of Information and Communication Technology, Swinburne University of Technology, Australia, Technical Report 1-02 (2005)
2. Brownlee, J.: Clonal Selection Theory, ClonalG and The Clonal Selection Classification Algorithm (CSCA). CISCP, Faculty of Information and Communication Technology, Swinburne University of Technology, Australia, Technical Report 1-02 (2005)
3. Brownlee, J.: Immunos-81 The Misunderstood Artificial Immune System. CISCP, Faculty of Information and Communication Technology, Swinburne University of Technology, Australia, Technical Report 1-02 (2005)
4. Carter, J.H.: The Immune Systems as a Model for Pattern Recognition and Classification. Journal of the American Medical Informatics Association 7 (1) (2000)
5. Coakes, S.J., Steed, L.G.: SPSS Analysis without Anguish Version 11.0 for Windows. John Wiley and Sons Australia, Ltd (2003)
6. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
7. Dasarathy, Belur, V.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
8. Dasgupta, D.: Advances in Artificial Immune Systems. IEEE Computational Intelligence Magazine (2006)
9. de Castro, L.N., von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation 6(3), 239–251 (2002)
10. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
11. Freitas, A., Timmis, J.: Revisiting the Foundations of Artificial Immune Systems for Data Mining. IEEE Transactions on Evolutionary Computation 11(4) (2007)
12. Hamaker, J., Boggess, L.: Non-Euclidean Distance Measures in AIRS, an Artificial Immune Classification System. In: Proceedings of CEC 2004 (2004)
13. Hart, E., Ross, P.: Studies on the Implications of Shape-Space Models for Idiotypic Networks. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 413–426. Springer, Heidelberg (2004)
14. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory 14, 515–516 (1968)
15. Hunt, J.E., Cooke, D.E.: Learning Using an Artificial Immune System. Journal of Network Computer Applications 19, 189–212 (1996)

16. Keogh, E.: UCR Time Series Data Mining Archive. University of California, Riverside, USA (2006), http://www.cs.ucr.edu/~eamonn/TSDMA/index.html
17. Merz, C.J., Murphy, P.M.: UCI Machine Learning Repository. University of California, Irvine, USA (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
18. Stanfill, C., Waltz, D.: Towards Memory-based Reasoning. ACM 29, 1213–1228 (1986)
19. Timmis, J.: Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory. Ph. D. thesis, Department of Computer Science, University of Wales, Aberystwyth (2001)
20. Timmis, J.: Challenges for Artificial Immune System. In: Apolloni, B., Marinaro, M., Nicosia, G., Tagliaferri, R. (eds.) WIRN 2005 and NAIS 2005. LNCS, vol. 3931, pp. 355–367. Springer, Heidelberg (2006)
21. Ventura, Dan, Martinez, T.R.: An Empirical Comparison of Discretization Methods. In: Proceedings of the 10th International Symposium on Computer and Information Sciences, pp. 443-450 (1995)
22. Watkins, A.: A Resource Limited Artificial Immune Classifier. Master's Thesis, Mississippi State University (2001)
23. Watkins, A., Timmis, J., Boggess, L.: Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. Genetic Programming and Evolvable Machines 5, 291–317 (2004)
24. Watkins, A., Timmis, A.: Artificial Immune Recognition System (AIRS): Revisions and Refinements. In: Proceeding of ICARIS 2002, pp. 173–181. Springer, Heidelberg (2002)
25. Watkins, A., Timmis, J.: Exploiting the Parallelism Inherent in AIRS, an Artificial Immune Classifier. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 427–438. Springer, Heidelberg (2004)
26. Wilson, D.R., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
27. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers, San Francisco (2005)
28. Zwitter, M., Milan Soklic, M.: Institute of Oncology, University Medical Center, Ljubljana, Yugoslavia (1998)

An Artificial Immune System for Evolving Amino Acid Clusters Tailored to Protein Function Prediction

A. Secker[1], M.N. Davies[2], A.A. Freitas[1], J. Timmis[3], E. Clark[3], and D.R. Flower[2]

[1] Computing Laboratory and Centre for BioMedical Informatics, University of Kent, Canterbury, CT2 7NF, UK
[2] The Jenner Institute, University of Oxford, Compton, Newbury, Berkshire, RG20 7NN, UK
[3] Departments of Computer Science and Electronics, University of York, York, YO10 5DD, UK
a.d.secker@kent.ac.uk, m.davies@mail.cryst.bbk.ac.uk, a.a.freitas@kent.ac.uk, jtimmis@cs.york.ac.uk, edclark@cs.york.ac.uk, darren.flower@jenner.ac.uk

Abstract. This paper addresses the classification task of data mining (a form of supervised learning) in the context of an important bioinformatics problem, namely the prediction of protein functions. This problem is cast as a hierarchical classification problem, where the protein functions to be predicted correspond to classes that are arranged in a hierarchical structure, in the form of a class tree. The main contribution of this paper is to propose a new Artificial Immune System that creates a new representation for proteins, in order to maximize the predictive accuracy of a hierarchical classification algorithm applied to the corresponding protein function prediction problem.

Keywords: artificial immune systems, data mining, bioinformatics, classification, clustering.

1 Introduction

This paper addresses classification within data mining in the context of bioinformatics, more precisely the prediction of protein function. In essence, a protein consists of a linear sequence of amino acids, and predicting the function of a protein, based on information derived from its sequence of amino acids, remains an important problem in bioinformatics.

The main contribution of this paper is to propose a new Artificial Immune System (AIS) – a variant of opt-aiNet (a well-known AIS) – that creates a new representation for proteins, in order to maximize the predictive accuracy of a classification algorithm applied to the corresponding protein function prediction problem.

In order to understand the task to be solved by the proposed AIS, it should first be noted that the type of attribute representation addressed in this paper involves local descriptors of amino acid sequences [18], [7]. In developing the local descriptors technique, Cui et al. [7] divided the amino acids into three functional groups (clusters); namely hydrophobic, neutral and polar, based upon the amino acid clustering suggested by Chothia and Finkelstein [5]. There are, however, many different ways of

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 242–253, 2008.

clustering amino acids, according to many different physical-chemical properties. Furthermore, it is unlikely that a given amino acid clustering will be the most effective one for all possible protein function prediction problems. The optimal amino acid clustering tends to be strongly dependent on the type of protein being classified.

In this context, this paper proposes an AIS that evolves clusters of amino acids optimized for a given type of protein. The evolved clusters are then used to define the protein representation that will be used by the classification algorithm. In the words of machine learning and data mining, the AIS algorithm solves a clustering (unsupervised learning) problem, consisting of finding the optimal clustering of amino acids for the type of protein whose data is being mined, and the result of the AIS is then used to solve a classification (supervised learning problem).

The proposed AIS is evaluated on a challenging real-world protein function prediction problem: the classification of GPCRs (G-protein-coupled receptors) into their functional classes. GPCRs constitute a large and diverse group of proteins that perform many important physiological functions [6], [12], [3]. The addressed GPCR classification problem is challenging because it involves a large number of classes organized in a hierarchy – being an instance of the so-called hierarchical classification problem – as will be explained later.

The remainder of this paper is organized as follows. Section 2 describes how the problem of predicting GPCR functions is cast into a classification problem. This section also provides some background on bioinformatics, in order to make the paper more understandable to readers without a biology background. Section 3 described the proposed AIS for clustering amino acids. Section 4 reports computational results, and Section 5 concludes the paper.

2 Casting Protein Function Prediction as a Classification Problem in Machine Learning/Data Mining

2.1 Representing Proteins by Local Descriptors of Amino Acid Sequences

Proteins are large molecules that perform a wide range of vital functions in living organisms. A protein consists of a linear sequence of amino acids – each of which can be represented by a single letter. For instance, the sub-sequence "AVC…" corresponds to (A)lanine, (V)aline, (C)ysteine, … Given a protein's sequence of amino acids, one can try to determine its function via either biological experiments or computational prediction methods. The former produce in general more precise results, but are much more time consuming and expensive. Hence, the latter is often used in practice, and it can provide valuable information for the more cost-effective use of biological experiments. This work addresses the computational prediction of protein function, by casting this problem as a classification (supervised learning) problem in machine learning/data mining, where protein functions are classes and attributes derived from the protein's sequence of amino acids are the predictor attributes.

The number of amino acids in the sequence varies widely across different proteins. Since the vast majority of classification algorithms can cope only with datasets where all examples (records, data items) have the same length, it is necessary to convert all proteins (examples) to the same fixed number of attributes, using an attribute

representation at higher level of abstraction than the full sequence of amino acids. The high-level representation used here involves the attribute creation technique defined in [18], which is based on summarizing the protein's entire sequence of amino acids by a fixed number of local descriptors (attributes), as follows.

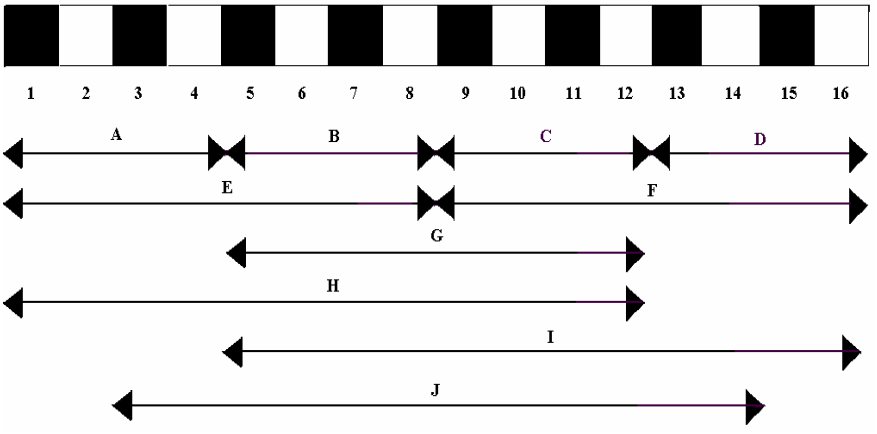


Fig. 1. The 10 descriptor regions (A-J) for a hypothetical protein sequence of 16 amino acids. Adapted from Zhang *et al.* [18] (unpublished).

Cui et al. [7] divided the amino acids into three functional clusters: hydrophobic (amino acids C,V,L,I,M,F,W), neutral (amino acids G,A,S,T,P,H,Y), and polar (amino acids R,K,E,D,Q,N), as suggested by Chothia and Finkelstein [5]. It is then possible to substitute the amino acids in the sequence for the cluster in which that amino acid belongs. Assuming H=hydrophobic, N=neutral and P=polar, the protein sequence CVGRK would be converted to HHNRR. The position or variation of these clusters within a sequence is the basis of three local descriptors: composition (C), transition (T), and distribution (D).

C is the proportion of amino acids with a particular property (drawn from a particular cluster such as the hydrophobic one). As an example, given the cluster H, we can determine C(H) over the example sequence of HHNRR as 0.4 as 2 of 5 positions in the sequence are of value H. T is the frequency with which amino acids with one property are followed by amino acids with a different property. Thus to compute T(N) over the example sequence, we can see there is a transition between H and N from positions 2 to 3, then a transition from N to R between positions 3 and 4. In this case T(N) = 2/4 = 0.5 as there are 4 places where a transition may occur. Any transitions between H and R are ignored here as neither of these clusters are the subject. Descriptor D measures the chain length within which the first, 25%, 50%, 75% and 100% occurrences of the particular property are located.

Given that the amino acids are divided into three clusters in this instance, the calculation of the C, T and D descriptors generates 21 attributes in total (3 for C, 3 for T and 15 for D). While this technique is valid if applied over the whole amino acid sequence, Zhang et al. [18] split the amino acid sequences into 10 overlapping regions – see Fig 1. For sequences A-D and E-F there may be cases where the

sequence cannot be divided exactly, in which case each subsequence may be extended by one residue. Each descriptor - C, T, and D - is calculated over the 10 subsequences. The number of attributes created with this technique therefore generalises to $70n$, where n is the number of amino acid clusters. In the case of 3 clusters of amino acids, proteins are now represented by 210 numerical descriptors, which can be offered to any of the plethora of well understood, well documented classification algorithms.

In the case of [18], the three clusters as defined in [7] were used, however no explicit explanation was included to justify the use of this particular clustering scheme. The keys to the success or failure of the technique described thus far are: (a) the number of clusters used, and (b) the specific amino acids that are included in each cluster. While there exists a truly enormous number of ways to partition the 20 amino acids, it seems clear that some will be more useful than others. However, in general it is not possible to determine, *a priori*, which amino acid clustering will result in the optimal performance for a given protein dataset. In addition, the classifier used may have certain biases that can be exploited during the clustering procedure. Hence, in principle we can use a data-driven approach to evolve an amino acid clustering that approaches optimality with respect to both the data being mined and the classification algorithm applied to that data.

This is the approach followed in this paper, whose main contribution is to present a new variant of the opt-aiNet algorithm for producing an amino acid clustering tailored to the problem of protein function prediction – cast as a classification problem.

2.2 Hierarchical Classification of G-Protein-Coupled Receptors (GPCRs)

Some data can be naturally organised as a hierarchy of classes. The classification of data in such a hierarchy poses some unique challenges to data miners, such as the large number of classes to be predicted. One particular case of this is the classification of G-Protein Coupled Receptor (GPCR) proteins by their function. GPCRs are important proteins as they can transmit messages from a cell's exterior to its interior, changing that cell's behaviour, and approximately 50% of all marketed drugs are targeted towards GPCRs [13].

The method of optimising clusters for a local descriptor-based attribute construction technique, as proposed in this paper, is generic to any protein dataset where it is sensible to represent the data using the local descriptors representation, but it should be pointed out that the GPCR dataset used in this study is hierarchical in nature. Because of this, the algorithm used to assess the quality of the attriute-construction technique and compare it with a baseline is also hierarchical in nature. Most extant classifiers deal with flat data sets, i.e., data for which a single level of classes may be assigned to an example. In a hierarchical dataset an example may be assigned to one class at a number of levels of specialisation. The most general level being near the root of the tree and becoming more specialised as the tree's branches are traversed. In this paper we deal only with structures where each class has exactly 1 parent – i.e. the data is structured like a tree. The class structure of a typical flat dataset will contain, for example, classes A, B and C which are all equally different from each other. However, in a hierarchy some classes may be more alike than others. Classes A and B are equally dissimilar, but these classes may subdivide such that classes A1 and A2

are more alike than A1 and B1 as A1 and A2 share a common parent class. For more details about the hierarchical classification of GPCRs, see [14].

3 The Proposed Artificial Immune System for Amino Acid Clustering

Pseudocode 1 shows the most general view of the process of attribute creation based on amino acid clustering (performed by an AIS) and subsequent use of a classification algorithm. Note that this process of attribute *creation* (or *construction*) based on clustering should not be confused with attribute *selection*. The goal of attribute selection is essentially to choose a subset of relevant attributes, out of all available attributes. This work rather involves attribute construction, where the goal is to create new attributes (new descriptors of amino acid sequences corresponding to higher-level information about proteins) based on the original sequence of amino acids (corresponding to lower-level information about proteins). The actual process of attribute creation is performed by using a clustering algorithm that groups together similar amino acids, and the result of this clustering is then used to produce a new set of predictor attributes for the classification algorithm.

```
1. Split full dataset into training and testing sets
2. Split training set into sub-training and validation sets
3. Generate initial random candidate clustering solutions
4. Evolve clustering
   4a. Create attributes for sub-training and validation
       data from clusters
   4b. Train classifier on sub-training data
   4c. Evaluate classifier on validation data
   4d. Assign quality to this clustering
   4e. Update population depending on individual's quality
   4f. Repeat from 4 until stopping criterion is met
5. Return the best clustering from the population
6. Create attributes for training and testing datasets using
   this best clustering
7. Train classifier using newly transformed training set
8. Evaluate classifier using newly transformed test set.
```

Pseudocode 1. High level description of amino acid clustering-based attribute creation and subsequent use of classification algorithm

In Pseudocode 1, points 1 and 2 are standard pre-processing tasks. Point 3 initialises the population for the AIS that performs amino acid clustering; while point 4 and sub-points thereof describe, at a high level of abstraction, the evolutionary process of amino acid clustering. Point 6 uses the output of the AIS (point 5) to create the data which will form the input to the classification algorithm, while points 7 and 8 are the standard training/testing steps used in a classification scenario.

The proposed AIS for amino acid clustering is a new variant of opt-aiNet, which we call opt-aiNet-AA-Clust (opt-aiNet for Amino Acid Clustering). The original opt-aiNet is an optimiser based on abstract ideas of clonal selection and somatic hypermutation as found in natural immune systems [11]. Opt-aiNet was first proposed in [9],

[10], and updated slightly in [16]. In this latter paper, opt-aiNet was proposed as a function optimisation tool. In this case, each immune cell would encode a single floating point value – the input to the function to be optimised.

Several modifications were required to allow the opt-aiNet algorithm to work in our scenario of amino acid clustering. These included the changing of the individual representation from a real value to a string of symbols to represent clusters, the changing of the fitness evaluation from a straightforward mathematical function to a much more complex system for creating and evaluating the attributes produced by the clustering results and some minor procedural changes such as the termination function. In the case of the original opt-aiNet, the algorithm will terminate when there has been no improvement above a threshold in the population between successive iterations. In this case, it is possible that many iterations could pass before an improvement is found and thus the system terminates after a given number of iterations. These changes are explained in more detail below.

Individual (Immune Cell) Representation. Each individual (immune cell) encodes a candidate solution to the problem of clustering the 20 amino acids. More precisely, each individual consists of a vector with 20 elements, $<c_1, \ldots, c_{20}>$, where the *i*th element, $c_i = 1,..,20$, indicates the id of the cluster to which the ith amino acid is assigned – since there are 20 amino acids. To consider a simple hypothetical example, if the first five elements of a vector were 3, 1, 2, 1, 3, this would mean that the second and fourth amino acids would be assigned to the same cluster (arbitrarily denoted as cluster 1); the first and fifth amino acids would be assigned to another cluster (denoted as cluster 3); and the third amino acid would be assigned to yet another cluster (denoted as cluster 2); and so on, for all the 20 amino acids. Different individuals can produce different numbers of clusters.

The Algorithm's Pseudocode and Search Operators. The opt-aiNet-AA-Clust algorithm proceeds as shown in Pseudocode 2, which is a more detailed description of points 4a-4f from Pseudocode 1. Thus, the algorithm is initialised by generating a population of immune cells such that the representation of each immune cell is in a random configuration. That is, amino acids are randomly assigned to clusters. Next, the quality of each immune cell (that is, the accuracy of the attributes defined by the clustering represented by that individual) is assessed. This is a somewhat complex process, explained in the Fitness Function paragraph. Each immune cell is then cloned (copies of that cell are produced) mimicking the clonal expansion stage of an immune reaction. These clones are mutated with a rate inversely proportional to their parent's (and therefore their) quality. The mutation scheme used in this algorithm is somewhat different to the original opt-aiNet. In the latter, the single value encoded by each immune cell will be incremented or decremented with a magnitude based on its fitness. However, a mutation in this context is simply a change in one or more positions in the immune cell's representation. This has the effect of switching an amino acid from one cluster to another. As well as switching an amino acid between clusters, this would include taking the amino acid out of a cluster with others and placing it in a cluster on its own or vice versa. The better the solution encoded by an immune cell the fewer positions are mutated. This has the effect of drastically changing poorly performing clustering schemes in the hope that a better solution may be found, while at the same time not destroying solutions that are already good. These newly mutated clones are

then assessed for quality once again and the best solution is kept to form part of the next generation. When all immune cells in the population have been cloned and mutated a small number of badly performing cells are discarded. These are replaced in the population with an equal number of randomly configured immune cells. This injection of randomness into the population discourages the population converging prematurely on a single local optimum.

```
1.Initalise population with each cell having randomly generated
   features
2. While (stopping criteria not met)
   2a. Determine fitness of each cell
   2b. Generate clones for each cell, keeping the parent cell in
       the population
   2c. Mutate each clone based on the fitness of its parent
   2d. Determine the fitness of all new clones
   2e. For each parent cell, select its fittest clone for sur-
       vival into next generation
   2f. Determine average fitness of the population. If it has
       improved significantly, then loop from 2.
   2g. Remove the least fit cells from the population
   2h. Replace the cells removed in 2.g. with randomly generated
       new cells
```

Pseudocode 2. opt-aiNet (adapted from [10])

Fitness Function. The original opt-aiNet used a single mathematical function as a measure of quality whereas the assessment of quality for each immune cell in this scenario is not as straightforward. Several stages must be gone through to assess the quality of the representation as encoded by the immune cell. For each immune cell, the clustering must firstly be translated from the immune cell representation, as explained earlier. The clusters defined can then be used to create a set of predictor attributes. In detail, each protein sequence in the training data set is split into 10 regions as defined in Fig. 1. Then the C, T and D (Composition, Transition and Distribution) values are determined for each protein subsequence (A-J) based on the clusters defined by the immune cell. This produces a dataset consisting of 70n predictor attributes (where n is the number of clusters as defined by the immune cell). This dataset (the training data) must then be split into two further sets – sub-training and validation. For this algorithm the split between these datasets is 80%/20%. The chosen classification algorithm is now trained on the sub training data and evaluated using the validation data. The quality of the cell's representation is defined as the percentage predictive accuracy output from the classifier on the validation set. Note that this predictive accuracy is measured on the validation set, separated from the sub training set (used to build a classification model), because the goal is to estimate the generalization ability of classification models, as is usual in classification.

Parallel processing. As each immune cell encodes a different set of clusters, it is important to note here that the above-described entire process of creating the new training set from the encoded clustering and then training/evaluating the classifier

must be repeated every time a fitness evaluation is requested and each iteration of opt-aiNet-AA-Clust may require many hundreds of such evaluations to occur. The fitness evaluation in this AIS is therefore extremely processor-intensive and as such the assessment of immune cell fitness was distributed over a cluster of 30 computers. Given each node in the cluster has its own copy of the training partition of the data set, each fitness evaluation is atomic in nature. Therefore multiple fitness evaluations can occur simultaneously while the algorithm pauses until all evaluations are complete. The main algorithm can then resume and continue as if the fitness evaluations had taken place in the normal, serial manner. It was found that executing these fitness evaluations in parallel was the only way to ensure the algorithm completed a reasonable number of iterations in a reasonable amount of time.

4 Computational Results

The new variant of opt-aiNet proposed in Section 3 – opt-aiNet-AA-Clust – was implemented by modifying the original opt-aiNet's code kindly obtained from Andrews [1], which formed part of [2]. The WEKA data mining toolkit [17] was used to provide the classification algorithm used in the fitness function, many of the algorithms used in the selective top-down classifier and a number of auxiliary functions regarding data manipulation. Some algorithms from [4] were also used in the selective top down classifier. The dataset used for training and testing was our own comprehensive dataset of GPCR sequences. This dataset, called the GDS dataset, originally contained 8354 protein sequences (examples), but classes with fewer than 10 examples were discarded – since in general such rare classes cannot be reliably predicted. This left 8222 protein sequences in the dataset. The dataset contains 5 classes (A-E) at the family level (the first level), 40 classes at the sub-family level and 108 classes at the sub-sub-family level (the third level). This dataset is described in more detail in [8].

For each run of opt-aiNet-AA-Clust, the algorithm was run on the training data and then the classification algorithm was trained on the same training data. Hence, following standard machine learning principles, no data used during the amino acid clustering stage was present in the ultimate testing of the classifier. For each run of the algorithm the number of training items was reduced to half the size by random sampling, in order to reduce processing time – due to the rather processor-intensive fitness function.

Ideally, the opt-aiNet-AA-Clust's fitness function would use a classification algorithm to predict classes in all 3 hierarchical levels of GPCR function. However, this is prohibitively slow with each individual evaluation likely to take many hours. Clearly a faster solution must be found. It was decided that just one classifier should be used in the fitness function. As 1-Nearest Neighbour (1-NN) has appeared to be the more accurate than other classifiers on this type of data in preliminary tests, it was chosen here. As only one classifier is to be used, it was decided that for the purpose of fitness computation the classifier will distinguish between classes only at the top level of the hierarchy (GPCR families A-E).

For each opt-aiNet-AA-Clust run, the algorithm performs 40 generations, using a population size of 20 individuals. While the algorithm was allowed to form clusters

using any combination of amino acids, a limit of 5 clusters per individual was enforced. Because of the way the clustering is used to produce the predictor attributes, large number of clusters per individual results in a very large number of predictor attributes, and so the classifier becomes too slow to train and test in a reasonable amount of time. Thus, it was decided that 5 clusters struck a reasonable balance between the algorithm's flexibility and constraining the time taken during evaluation of the representation. Table 1 shows the parameters used for each run of opt-aiNet-AA-Clust.

Table 1. Opt-aiNet-AA-Clust parameters

Number of initial cells in the network	20
Number of clones for each immune cell during clonal selection	20
Number of algorithm iterations	40
Suppression threshold for network cell affinities	0.5
Maximum number of clusters that can be produced by each immune cell	5
Fitness evaluation method	1-NN classifier

To assess the effectiveness of the proposed algorithm, an experiment was undertaken to compare the accuracy of a classifier when attributes are evolved by the algorithm against a baseline. As stated above, the dataset used was our GDS dataset. In the case of the baseline, attributes were generated from raw protein sequences by the approach of Zhang et al. [18], as described earlier. For each set of constructed attributes the same classification algorithm was used. In this case it was the selective top down classification algorithm as defined in [8] and [15]. In other words, the experiments compare the performance of a given hierarchical classification method in two different scenarios, using two different types of predictor attributes: the attributes created by using our proposed opt-aiNet-AA-Clust and the baseline attributes proposed by Zhang et al. [18]. Hence, what is ultimately being compared is the effectiveness of two different protein representations: one of them automatically evolved by opt-aiNet-AA-Clust and the other manually proposed by Zhang et al. using their domain knowledge about proteins and amino acid properties.

Because of the sheer amount of time taken to evolve the protein representations, only one run of a 10-fold cross-validation procedure – a standard procedure for evaluating predictive accuracy in data mining [17] – was performed with opt-aiNet-AA-Clust. However, as the experiments with the baseline representation have been run before during other investigations, the results of 10 runs of a 10-fold cross-validation procedure (100 runs of the classifier in total) are available. The results are shown in Table 2 where the mean predictive accuracy over the 10 folds of the cross-validation procedure is shown. The mean accuracies for the baseline are shown and finally the statistical significance of the difference between the accuracies of the evolved representation and the baseline is displayed. This has been computed using Student's t-test with 2-tails. This test was used as the number of runs is small while it can be used to compare distributions where there are different numbers of observations for each. In this case, 10 observations for the evolved attributes and 100 for the baseline.

Table 2. Predictive accuracy (%) per class level

	1st level	2nd level	3rd level
Classifier using attributes evolved by opt-aiNet-AA-Clust	96.91	83.14	72.75
Classifier using baseline attributes	96.97	82.72	70.46
P value result of Student's t-test	0.775	0.280	0.003

It can be seen from the table that the difference in the predictive accuracy of the two approaches on the first (most general) and second class levels are statistically negligible – the t-tests produced high p values. On the other hand, at the third class level the attributes evolved by opt-aiNet-AA-Clust led to a very significant improvement in predictive accuracy over the baseline attributes, statistically significant at the 1% level.

It should be noted that the third class level represents the most challenging classification scenario, since it involves many classes and typically a smaller number of examples per class (making generalization more difficult), as compared with the first two levels. In addition, classes at the third level are often more informative to biologist users, since they specify a protein's function more precisely.

It should be stressed that, although the automatically evolved clusters of amino acids have led to an improvement for the particular dataset of GPCR proteins used in our experiments, there is no guarantee that the same evolved amino acid clusters will be optimal for predicting other types of protein functions. However, the proposed algorithm is generic enough to be easily applicable to other types of proteins, offering us an automated approach for trying to find a near-optimal cluster of amino acids tailored to the type of protein whose functions have to be predicted.

5 Conclusions

Previous experience has shown that the protein representation generated by the local descriptors method results in highly competitive predictive accuracies when attempting to classify GPCR proteins. The local descriptors technique, as currently published in the literature, divides amino acids into 3 clusters, leading to a specific set of predictor attributes. When evaluating this published representation, , we found no clear reason why these three clusters were used. It was therefore hypothesised that predictive accuracy could be improved over this "one size fits all" set of clusters by assigning amino acids to clusters in a data driven manner. In this spirit, this paper proposed a new variant of opt-aiNet, called opt-aiNet-AA-Clust, that optimizes the clustering of amino acids for the type of protein being mined and for the type of classification algorithm being used.

When compared against the original local descriptors-based representation, which was not optimized for the data nor for the classification algorithm, it was found that a significant increase in predictive accuracy was observed at the 3rd level of the class hierarchy, which is the most informative (most specialized) type of protein function for the user.

One future direction would be to let the AIS algorithm have free reign to decide the number of clusters. It is thought that allowing an unlimited number of clusters could result in better predictive accuracy. However, in the experiments reported here this was impractical as, firstly, the AIS would have a hugely increased solution space to search, which would require an increase in time taken to solve the clustering problem. Secondly, an increase in the number of clusters defined by the solution returned by the AIS would result in a huge number of attributes being created for the data, which can be impractical when using a hierarchical classification algorithm.

References

1. Andrews, P.: opt-aiNet source code in Java, last modified October 2005 (Personal communication, 10 July 2007)
2. Andrews, P.S., Timmis, J.: On Diversity and Artificial Immune Systems: Incorporating a Diversity Operator into aiNet. In: Apolloni, B., Marinaro, M., Nicosia, G., Tagliaferri, R. (eds.) WIRN 2005 and NAIS 2005. LNCS, vol. 3931, pp. 293–306. Springer, Heidelberg (2006)
3. Bissantz, C.: Conformational changes of G protein-coupled receptors during their activation by agonist binding. J. Recept. Signal. Transduct. Res. 23, 123–153 (2003)
4. Brownlee, J. WEKA Classification Algorithms. Version 1.6. (retrieved December 2006), http://sourceforge.net/projects/wekaclassalgos (2006)
5. Chothia, C., Finkelstein, A.V.: The Classification and Origins of Protein Folding Patterns. Annual Review of Biochemistry 59, 1007–1035 (1990)
6. Christopoulos, A., Kenakin, T.G.: Protein-coupled receptor allosterism and complexing. Pharmacology Review 54, 323–374 (2002)
7. Cui, J., Han, L.Y., Li, H., Ung, C.Y., Tang, Z.Q., Zheng, C.J., Cao, Z.W., Chen, Y.Z.: Computer prediction of allergen proteins from sequence-derived protein structural and physicochemical properties. Mollecular Immunology 44, 514–520 (2007)
8. Davies, M.N., Secker, A., Freitas, A.A., Mendao, M., Timmis, J., Flower, D.R.: On the hierarchical classification of G Protein-Coupled Receptors. Bioinformatics 23(23), 3113–3118 (2007)
9. de Castro, L., Von Zuben, F.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems 6(3), 239–251 (2001)
10. de Castro, L.N., Timmis, J.: An artificial immune network for multimodal optimisation. In: Congress on Evolutionary Computation (CEC 2002). Part of the 2002 IEEE World Congress on Computational Intelligence, Honolulu, Hawaii, USA, pp. 699–704. IEEE, Los Alamitos (2002)
11. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
12. Gether, U., Asmar, F., Meinild, A.K., Rasmussen, S.G.: Structural basis for activation of G-protein-coupled receptors. Pharmacological Toxicology 91, 304–312 (2002)
13. Klabunde, T., Hessler, G.: Drug Design Strategies for Targeting G-Protein Coupled Receptors. Chem. Bio. Chem. 3, 928–944 (2002)
14. Secker, A., Davies, M.N., Freitas, A.A., Timmis, J., Mendao, M., Flower, D.R.: An Experimental Comparison of Classification Algorithms for the Hierarchical Prediction of Protein Function. Expert Update (Magazine of the British Computer Society's Specialist Group on AI), Special Issue on the 3rd UK KDD (Knowledge Discovery and Data Mining) Symposium 9(3), 17–22 (2007)

15. Secker, A., Davies, M.N., Freitas, A.A., Timmis, J., Mendao, M., Flower, D.R.: An Experimental Comparison of Classification Algorithms for the Hierarchical Prediction of Protein Function. In: 3rd UK Data mining and Knowledge Discovery Symposium (UKKDD 2007), Canterbury, pp. 13–18 (2007)
16. Timmis, J., Edmonds, C.: A Comment on opt-AINet: An Immune Network Algorithm for Optimisation. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 308–317. Springer, Heidelberg (2004)
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
18. Zhang, Z.H., Tammi, M.T., Zhang, G.L., Tong, J.C.: Prediction of protein allergenicity using local description of amino acid sequence (unpublished) (2005)

Optimization of Steel Catenary Risers for Offshore Oil Production Using Artificial Immune System

Ian N. Vieira[1], Beatriz S.L.P. de Lima[1,2], and Breno P. Jacob[1]

[1] COPPE and [2] Polytechnic School /Federal University of Rio de Janeiro
P.O.Box 68506, 21941-972 - Rio de Janeiro, Brazil

Abstract. This work presents an application of Artificial Immune System (AIS) using CLONALG to the synthesis and optimization procedure of a Steel Catenary Riser (SCR) for floating oil production systems at deep and ultra-deep waters. The evaluation of the behavior of riser configurations, needed for the calculation of the fitness function in the optimization procedure by an evolutionary algorithm, requires a large number of time-consuming Finite Element analyses. Therefore, it is important to reduce the number of analyses; in this paper, the effectiveness of AIS for this purpose is assessed in this real-world industrial application. The results indicate that the AIS approach is more effective than Genetic Algorithms (GA), generating better solutions with smaller number of evaluations.

Keywords: Artificial immune System, Optimization, Steel Catenary Risers.

1 Introduction

Developing oil production activities in deep and ultra-deep waters has been a permanent challenge for petroleum industries around the world. The design of production risers, to connect the wellheads at the seabottom with a floating platform at the sea surface, is one crucial aspect of these activities.

For water depths below 1000 m, flexible risers have been employed with most floating production systems. However, for ultra-deep waters flexible risers can frequently reach or exceed their technical and economical feasibility limits. This fact can occur due to the limitation of the viable riser diameter, particularly when associated to high external pressures and temperatures, and to significant static offsets and heave motions, associated with reduced capability of sustaining harsh service conditions [1].

Recently, the SCR (Steel Catenary Riser) concept has been shown to be able to overcome such limitations, and to comprise a feasible alternative when employed in floating production systems. Detailed studies presented in [2] demonstrated that the *lazy-wave* configuration, in which distributed floaters are installed in an intermediate section of the riser (Fig. 1), presents a structural behavior more favorable than the usual free-hanging catenary, under environmental loadings of wave, marine current, and the motions imposed by the platform.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 254–265, 2008.

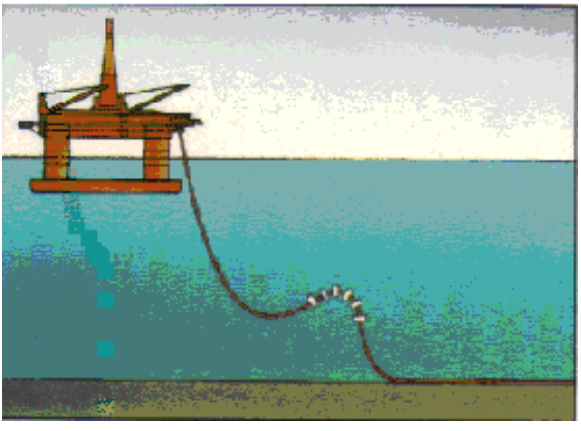


Fig. 1. Lazy-wave Configuration

Those studies included an exhaustive parametric analysis to achieve a better appreciation of the model behavior, and to determine values for some parameters that defines the configuration, including for instance top angle and length and position of the section with floaters, in order to obtain a configuration with reasonable costs and still capable of resisting extreme loadings and fatigue. However, those studies were performed with a particular scenario in mind, and the conclusions may not be readily extended for other scenarios, for instance in ultra-deep waters, where similar huge and expensive parametric studies would be necessary.

The motivation of the present work, therefore, lies in the recognition that the selection of a riser configuration with a good structural performance and low cost must indeed be formally described and treated as an optimization problem. This fact had already been recognized in [1], where Genetic Algorithms – GA were employed in the development of an optimization procedure for *lazy-wave* risers.

Nevertheless, the evaluation of the behavior of riser configurations, needed for the calculation of the fitness and objective function in the GA approach, requires Finite Element structural analyses employing a non-linear time-domain dynamic solver. Such solvers are extensively time consuming; furthermore, previous studies indicated that a typical optimization procedure using GA [1] could involve a large number of evaluations.

Therefore, an ideal optimization approach should be able to find an optimum solution in the shortest time possible, indicating that efforts should be directed to minimize the number of analyses for the calculation of the fitness of each candidate configuration. Therefore, the primary objective of this work is to minimize the number of evaluations by employing Artificial Immune System – AIS, comparing its performance with the optimization procedure employing GA.

This paper is organized as follows. Section 2 describes the optimization methodology applied in this work. Section 3 presents the formulation of the riser optimization problem. Section 4 describes a sensitivity study of some parameters of AIS and GA in the search for the lowest possible number of evaluations that can achieve an optimum solution. Finally, Section 5 presents the final remarks and conclusions.

2 Optimization Methodology

2.1 Engineering Optimization Problems

Real Engineering problems are usually difficult to model and complex by nature. Classical optimization approaches involves gradient information and requires a great computational effort [1]. Structural engineering optimization problems are characterized by various objectives and constraints functions that are generally non-linear functions of the design variables [3]. Conventional techniques may find difficulties such as being trapped in local optima, high dimensional spaces and multi-constraints, making the adoption of such techniques inappropriate.

AIS and other nature-inspired computation paradigms, such as GA, artificial neural networks and applications of swarm intelligence (such as particle swarm and ant colony optimization), have been shown to be efficient and robust at solving complex problems in engineering. Some previous works present the application of GA to structural engineering optimization problems. Initial applications of GA to optimize structural topologies were carried out by Shankar and Hajela [4] and Hajela *et al* [5]. Degertekin [6] presents the optimum design of geometrically non-linear elastic-plastic steel frames using GA.

The computational cost of fitness function evaluation can be a serious dilemma to GA applied to structural optimization problems. Some works present techniques that aim to reduce the computational effort of such algorithms. In [7] a new Evolutionary Algorithm (EA) is introduced with a globally stochastic but locally heuristic search strategy by incorporating a modified micro-genetic algorithm with two local optimization operators. The authors state that the optimal design is obtained with fewer computational operations than the ones required by the existing algorithms. Fonseca et al [8] employ approximation models to replace the objective function evaluations; essentially, that approach consist in the correct evaluation of only a fraction of the population. The results showed that the approximation procedures could be embedded in a GA, allowing computational gains.

Yoo and Hajela [9] made a first trial of using immune system inspiration to solve structural optimization problems. It was based on the ability of the immune system of performing pattern recognition to improve the performance of the GA in solving a multicriterion design problem. Luh *et al* [10], describes a novel approach for finding optimal solutions to multi-modal structural problems using a multi-modal immune algorithm (MMIA). The use of this approach provided this methodology with a superior local search ability compared to GA.

A deep discussion is brought in [11] about the role of AIS in many research areas, including examples of recent applications, and its potential in the future. It is clear that most of the publications related to optimization using AIS present results for benchmark cases rather than real-world problems. Specifically, structural engineering optimization problems using AIS are not very common, mainly because this approach has been proposed relatively recently compared to others EAs. This is exactly the purpose of this work: to implement and study an application of AIS to a real-world, complex structural optimization problem – the definition of an optimal *lazy-wave* SCR riser configuration for deepwater oil exploitation activities.

2.2 Artificial Immune System – CLONALG

AIS follows ideas taken from immunology in order to develop systems capable of performing different tasks in various areas of research. The clonal selection principle that is able to explain the basic features of an immune response to an antigen stimulus inspired the development of powerful computational tools [12].

The AIS considered in this work uses the Clonal Selection Algorithm – CLONALG [12] to implement the optimization procedure. This algorithm was initially proposed to carry out machine-learning and pattern recognition tasks, and then adapted to solve optimization problems.

Other AIS optimization algorithms could be used, such as AiNET; however CLONALG was chosen for our first approach in comparing GAs and AIS algorithms, because it is a basic and simpler algorithm, and mainly because, while AiNET has achieved better results in multimodal optimization, on average it uses more iterations to find the global optimum (as stated in [13]), indicating that it might require more affinity evaluations. Still according to [13], AiNET presents the advantage of locating a larger number of optima solutions; however, the objective of our current engineering application is not to obtain several optima solutions (in general only one solution will suffice), but to obtain the solution with the smaller number of fitness evaluations.

The basic CLONALG optimization algorithm may be written as follows:

1. Generate a random initial population of antibodies (*Ab)* of size *N*;
2. Calculate affinity values of the *Ab* population;
3. Generate *Nc* clones by cloning all *N* cells in the *Ab* population;
4. Mutate the clone population to produce a mature clone population;
5. Select the *N* best *Ab* to compose the new population;
6. Repeat steps 2-5 until it reaches a pre-defined stopping criteria.

The immune system uses a relatively small number of gene segments to construct the antibodies receptors. The rearrangement of those gene segments can generate an infinite number of antibodies that is able to cover antigenic spaces. There are four main classes of antibodies that can be switched during this rearrangement [14]. In this work, a computational procedure is inspired in a concept of different classes of antibodies, acting as "antibody families" that are able to co-evolve, Therefore, this idea is incorporated in the algorithm with evolutions taking place separately in each family.

The antibody family mechanism allows the algorithm to simultaneously explore different points of the search space. This characteristic, along with the affinity proportional mutation rate, provides the ability to search around every "family" according to its affinity. It is expected that this mechanism contributes to the aforementioned main objective of our current application: the reduction of the number of affinity/fitness evaluations when compared to GA.

In step 5, the selection operator is adapted to incorporate this concept. The best antibody is selected deterministically between the cells of each antibody family, by comparing each original antibody only with its clones. This mechanism can guarantee a good diversity during the evolution. Since we are interested in reducing the population size in order to reduce the number of function evaluations, this method allows the use of a very small number of antibodies without losing diversity.

The somatic hypermutation is the next event in the maturation of the immune cells, after the gene rearrangement, that can improve the affinity of the antibodies. The mutation rate applied to every immune cell is inversely proportional to its antigenic affinity. The mutation changes all attributes of an antibody vector according to the following expression [12]:

$$m' = m + \exp(-\rho \cdot D^*) \quad . \tag{1}$$

where $m = \langle m_1, m_2, ..., m_L \rangle$ is the attribute string, m' its mutated version, ρ is a parameter that control the smoothness of the inverse exponential, D^* is the normalized affinity, that can be determined by $D^* = D / D_{\max}$.

For real-world complex problems, the initial population should present a large number of cells in order to cover the complex landscape. In [12] the reproduction is made by cloning all antibodies of the population, giving rise to a temporary population of clones. The number of clones of each antibody is usually a percentage of the original population. Consequently, the temporary population can be very large as well as the number of function evaluations.

This characteristic is problematic to the present engineering problem, since our interest is in decreasing the number of evaluations. Therefore, the first step to reach this goal is to perform a sensitivity analysis varying the antibody population size. The results showed that, employing a small initial *Ab* population with a large number of clones, we could reach the desired reduction without losing diversity due to the cell family evolution.

3 Problem Representation

As mentioned in the introduction, the objective of this work is to develop a more efficient synthesis procedure for the determination of an optimal SCR riser system. This section describes the variables that are considered for the optimization process, employing real-valued shape-space for the representation of variables in the implementation of the AIS.

Fig. 2 presents a schematic model showing the parameters that define a *lazy-wave* riser system. The geometric riser parameters are (**L1**), length of lower riser segment; (**L2**), length of segments with distributed floaters; (**L3**), length of top segment of the riser; (α) the "top angle", or the angle of the riser axis with the vertical direction at the connection with the platform, measured in the neutral equilibrium configuration; (**Z**) the depth of the connection, and (**P**) the horizontal projection.

Since the horizontal projection **P** and the depth **Z** are dictated by the characteristics of the platform and well connections, and the angle α is related to the projection P and the total length (L1 + L2 + L3), only these latter geometric parameters need to be considered in the optimization process.

There are also the parameters related to the buoys, which are: (**Lf**) buoy length, (**Hdf**) buoy diameter, (**ESP**) spacing between buoys. Other parameters such as the specific weight and other mechanical properties of the buoys could be considered; however in this work only the geometric parameters Lf, Hdf and ESP will be optimized [15].

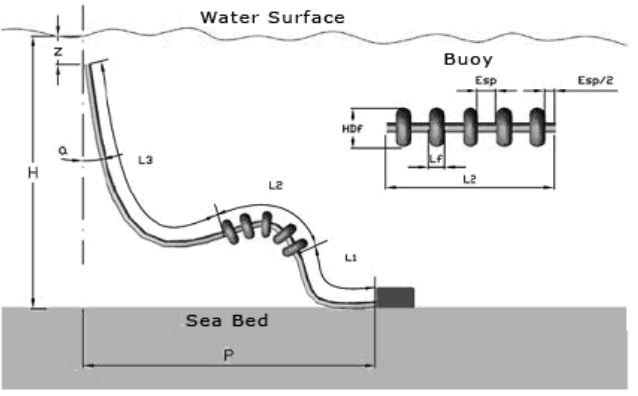


Fig. 2. Riser model

Therefore, there are six parameters to be optimized, in order to determine a riser configuration that complies with all technical standards and design criteria, and presents the lowest construction cost. For this purpose, the following cost function is used:

$$f = \frac{1}{f_{\max}} \cdot \left[\left(\sum_{i=1..n} CI_i \cdot L_i \right) + \left(V_{buoy} \cdot CI_{buoy} \right) \right] . \tag{2}$$

where $i = 1..n$ represents the number of segments of the riser; CI_i is the cost index associated to each segment; L_i is the segment length; CI_{buoy} represents the cost index associated to the volume of the buoy; V_{buoy} is the volume of the buoy and $f_{\max}$ is the maximum value of the objective function f .

Since this is a minimization problem with constraints, the affinity/fitness function will be defined as:

$$affinity / fitness = \left(f + \sum P_j \right)^{-1} . \tag{3}$$

where f is the objective function and $\sum P_j$ is the sum of all the penalties.

The structural behavior constraints are determined from the results of structural analyses. At this point, it is interesting to note that, although the final optimization procedure must rely in a full non-linear, time-domain dynamic Finite Element solver, in the current application the evaluations are performed using an analytical catenary solver, which is much faster to compute and provides results that are at least representative of the actual FE solution. This is because the main objective of this work is to assess the performance of the AIS approach in reducing the number of evaluations. In actual design applications, the evaluations will be performed by nonlinear dynamic FE analyses.

In any case, the structural behavior constraints are:

- The maximum equivalent *Von Mises* stress acting on the riser sections (to assure the structural integrity of the pipe);
- The maximum angle between the riser axis and the vertical direction at the connection with the platform (dictated by installation requirements);
- The maximum variation of the "built-in" angle, measured at the top riser axis, between the neutral equilibrium configuration and any configuration acquired by the riser during the application of the environmental loadings and the platform motions (dictated by the design of the *flex-joint* that provides an articulated connection of the riser with the platform);
- The maximum tension at the riser top (also dictated by the design of the *flex-joint*); and
- The minimum tension at the riser bottom (to avoid buckling and collapse of a riser section).

Therefore, there are five constraints, and the violation of any one of them results in a penalty given by the following equation:

$$P_j = \begin{cases} k \cdot (1 - x^3), \text{ if } x < 1 \\ 0, \text{ if } x \geq 1 \end{cases} . \tag{4}$$

where P_j is the penalty value of the $j-ith$ constraint criteria, x is the ratio between $j-ith$ constriction limit value and its calculated value and k is a factor to force the emergence of non restricted solutions.

4 Evaluation of the Performance of the AIS

4.1 Definition of the Structural Problem

This section presents results of test cases applying AIS and GA in the optimization of a given *lazy-wave* riser configuration, to be installed at a sea depth of 1290m, and considering a horizontal projection of 2000m. Specific data related to the riser modeling are depicted in Table 1. In this table, the cost ratio C2/C1 means that the segment with floaters costs two times more than the regular riser segments.

The user-defined bounds for the riser structural behavior constraints are displayed in Table 2.

Table 1. Riser modeling data

MATERIAL		GEOMETRY	
Density	7800 kg/m³	Thickness	0.01905 m
Specific weight	77 kN/m³	External diameter	0.21908 m
Yield stress	413 MPa	Internal diameter	0.18098 m
Allowable stress	277 MPa	Floater weight	0.162 ton/m
Elastic modulus	207800 MPa	Floater Buoyancy	0.3175 ton/m
Cost ratio C2/C1	2.0	Floater external diameter	0.568 m

Table 2. Riser optimization parameters

DESIGN LIMITS (meters)	MIN	MAX
Riser segment (L1)	800	2000
Riser segment (L2)	400	800
Riser segment (L3)	800	2000
Buoy diameter (HDf)	0.5	2
Buoy length (Lf)	0.5	2
Spacing between buoys (Esp)	0.8	1.5

CONSTRAINTS	VALUE
Von Mises Stress	415.5MPa
Maximum top angle	18°
Minimum top angle	5°
"Built-in" angle variation	5°
Maximum top Stress	1500 kN
Minimum Stress	300 kN

4.2 Definition of the AIS / GA Parameters

In order to assess the performance of the AIS in the minimization of the number of analyses for the affinity/fitness calculation, parametric studies are performed by establishing a set of values for the AIS and GA parameters. Of course, the success of the algorithm should be measured not only by the reduction of the number of fitness evaluations, but also by the average value of the best obtained fitness. Therefore, both results (the average of the best individual affinity/fitness and the average number of evaluations) will be taken to compare the results.

The current CLONALG implementation employs a family antibody concept in which every antibody compete only against its clones. Thus, a good diversity can be achieved naturally. Previous tests showed that after 50 generations no substantial improvement was attained in the affinity/fitness results.

Each line of Table 3 represents a test case for the AIS. All tests considered 50 generations, with the exception of the tests with population of 1000 individuals, which used 30 generations. Repeated individuals are not re-evaluated. Those latter tests with 1000 individuals are intended to provide a glimpse of the best possible solution for each algorithm.

In order to have a sensitivity of the hypermutation rate, each AIS test is executed with five different values of ρ (equation (1): 1, 2, 3, 4 and 5). For each value of ρ the test is repeated five times, therefore totalizing 425 executions of the algorithm.

Table 3. AIS Parametric Analysis – Test cases

TEST NUMBER	AB	CLONES	POP	TEST NUMBER	AB	CLONES	POP
AIS P 1	100	9	1000	AIS 8	8	9	80
AIS P 2	10	99	1000	AIS 9	6	9	60
AIS 1	10	9	100	AIS 10	4	9	40
AIS 2	10	8	90	AIS 11	2	9	20
AIS 3	10	7	80	AIS 12	2	19	40
AIS 4	10	6	70	AIS 13	2	29	60
AIS 5	10	5	60	AIS 14	2	39	80
AIS 6	10	4	50	AIS 15	2	49	100
AIS 7	10	3	40				

The classical Genetic Algorithm is used, with binary codification, single point crossover, one individual elitism and roulette-wheel selection. Four different mutation rates are employed: 0.05, 0.10, 0.15 and 0.20. The crossover rate is varied from 0.75 to 0.90 in steps of 0.05. Three different population sizes are considered: 1000, 100

and 80 individuals. The combination of these parameters results in 48 test sets for the GA. Each set is executed five times, thus totalizing 240 executions of the algorithm.

4.3 Results

Table 4 summarizes the results of the parametric analyses, comparing the performance of the GA and AIS algorithms in terms of statistical values (maximum, mean, minimum and standard deviation) for the affinity/fitness of the best individual obtained in the last generation of all executions of each test case.

The results are grouped according to the number of individuals. The first group includes the tests with the larger number of individuals (1000), and the second corresponds to all other tests with smaller populations. It can be observed that AIS is superior in all situations, achieving higher values of affinity.

Table 4. Summarized results of AIS and GA

	1000 INDIVIDUALS		SMALL POPULATION	
	AIS AFFINITY	GA FITNESS	AIS AFFINITY	GA FITNESS
MAX	1.635	1.540	1.630	1.507
MEAN	1.629	1.497	1.596	1.442
MIN	1.621	1.464	1.530	1.333
STD DEV	0.005	0.021	0.024	0.046

As mentioned before, the assessment of the performance of both methods should also include the number of fitness evaluations. Detailed results for each test case with the AIS are presented in Table 5, including the mean of the best antibody affinity/fitness (AFF) and the average number of evaluations (EVAL) amongst all executions for each test case.

It can be seen that the AIS was able to reach an affinity value of 1.621 with only 40 individuals (test case AIS 10, $\rho = 4$) requiring 1957 evaluations. Comparing with the maximum affinity of 1.635 that was obtained by a test with the larger population of 1000 individuals (and requiring around 30500 evaluations), it can be seen that this is a good result for such a small number of individuals. This complies with the main objective of this work, that is, to obtain accurate solutions with a small number of affinity/fitness evaluations.

In Table 5 one can also observe the influence of decreasing the number of antibodies: comparing the results of the test cases AIS 8 to 11, it can be seen that at the end of 50 generations around 1000 evaluations can be saved each time the number of antibodies is decreased by two. Also, comparing the results of the cases 1 to 7, it can be seen that a reduction of just one clone can save around 500 evaluations.

On the other hand, as can be seen in Table 4, the best fitness with the GA methodology obtained by a test with the larger population of 1000 individuals is smaller than the best fitness for the AIS. More important, the performance of the GA degraded remarkably with the reduction of the number of individuals. There was no need to try to reduce the population size below 80 individuals, since the performance is already unsatisfactory with 100 individuals.

Table 5. Detailed AIS results – mean of best individual affinity and number of evaluations

	ρ = 1		ρ = 2		ρ = 3		ρ = 4		ρ = 5	
TEST	**AFF**	**EVAL**	**AFF**	**EVAL**	**AFF**	**EVAL**	**AFF**	**EVAL**	**AFF**	**EVAL**
AIS P 1	1.621	30464	1.626	30893	1.629	30976	1.630	30988	1.622	31005
AIS P 2	1.625	30163	1.631	30362	1.634	30575	1.635	30537	1.635	30502
AIS 1	1.591	4894	1.610	4897	1.624	4901	1.610	4901	1.594	4902
AIS 2	1.603	4392	1.611	4411	1.620	4414	1.617	4411	1.618	4411
AIS 3	1.599	3913	1.619	3924	1.622	3922	1.596	3921	1.583	3924
AIS 4	1.606	3421	1.598	3434	1.613	3436	1.624	3436	1.551	3432
AIS 5	1.564	2938	1.585	2946	1.612	2947	1.601	2946	1.587	2945
AIS 6	1.589	2453	1.583	2458	1.588	2457	1.546	2457	1.545	2458
AIS 7	1.566	1965	1.575	1967	1.572	1968	1.567	1968	1.573	1969
AIS 8	1.599	3911	1.600	3920	**1.628**	**3922**	1.627	3918	1.596	3919
AIS 9	1.591	2939	1.608	2942	1.624	2940	**1.625**	**2940**	1.582	2940
AIS 10	1.585	1958	1.563	1962	1.585	1960	**1.621**	**1957**	1.582	1957
AIS 11	1.557	979	1.574	979	1.567	978	1.602	976	1.609	978
AIS 12	1.585	1952	1.597	1952	1.530	1954	1.585	1952	1.533	1949
AIS 13	1.591	2922	1.608	2923	1.605	2922	1.560	2912	1.614	2923
AIS 14	1.603	3885	1.615	3891	**1.626**	**3887**	1.625	3879	1.614	3885
AIS 15	1.607	4846	1.618	4848	**1.630**	**4860**	1.608	4849	1.626	4837

The best GA fitness obtained with 100 individuals is 1.507 (lower than any affinity value reported for the AIS in Table 5, which includes cases with a population of only 20 individuals). This result is achieved with an average of 2646 evaluations, more than most of the test cases with the AIS; for instance, more than the test case AIS 10, ρ = 4 that reached the affinity of 1.635 with an average of only 1957 evaluations.

5 Final Remarks

This work intends to study and implement two different nature-inspired optimization methodologies applied to a real-world, complex structural optimization problem – the definition of an optimal *lazy-wave* SCR riser configuration for deepwater oil exploitation activities. This is a demanding problem for the offshore industry, which requires cost-effective solutions for the exploration of oil in increasing water depths, in scenarios where both the cost and the complexity of the structures involved also tend to increase – therefore motivating studies on optimization procedures.

The evaluation of the behavior of riser configurations requires Finite Element structural analyses employing a non-linear time-domain dynamic solver. Such solvers are extensively time consuming; therefore, the primary objective of this work is perform sensitivity studies on some parameters of AIS and GA methodologies, searching for the lowest possible number of evaluations that can achieve an optimum solution.

In addition, some improvements are suggested in order to enhance the performance of the AIS, such as the concept of “antibody families” that co-evolve, in order to guarantee that a small population can reach a good diversity during the evolution.

The comparison between the results of this enhanced implementation of the AIS and those of a GA methodology showed that the former presented a superior behavior, being able to find an optimal solution with better affinity/fitness values while requiring fewer FE analyses to be performed.

The superior behavior of AIS against GA may be attributed to the concept of family selection, and also to the affinity-proportional mutation rate, which ensures the exploration of the search space with small population size.

Acknowledgements

The authors are grateful to the Brazilian research agency CNPq for the financial support of this research project.

References

1. Lima, B.S.L.P., Jacob, B.P., Ebecken, N.F.F.: A hybrid fuzzy/genetic algorithm for the design of offshore oil production risers. International Journal for numerical methods in engineering 64, 1459–1482 (2005)
2. Jacob, B.P., Lima, B.S.L.P., Reyes, M.C.T., Torres, A.L.F.L., Mourelle, M.M., Silva, R.M.C.: Alternative Configurations for Steel Catenary Risers for Turret-Moored FPSO's. In: Proceedings of the 9th International Offshore and Polar Engineering Conference, Brest-France, vol. 2, pp. 234–239 (1999)
3. Lagaros, N.D., Papadrakakis, M., Kokossalakis, G.: Structural optimization using evolutionary algorithms. Computers and Structures 80, 571–589 (2002)
4. Shankar, N., Hajela, P.: Heuristics driven strategies for near-optimal structural topology development. In: Topping, B.H.V. (ed.) Artificial intillegence structural engineering, pp. 219–226. Civil-Comp. Press, Oxford (1991)
5. Hajela, P., Lee, E., Lin, C.-Y.: Genetic algorithms in structural topology optimization. In: Bendsoe, M.P., Mota Soares, C.A. (eds.) Topology design of structures, pp. 117–133 (1993)
6. Degertekin, S.O., Saka, M.P., Hayalioglu, M.S.: Optimal Load and Resistance Factor Design of Non-Linear Steel Space Frames via Tabu Search and Genetic Algorithm. Engineering Structures 30, 197–205 (2008)
7. Fawaz, Z., Xu, Y.G., Behdinan, K.: Hybrid evolutionary algorithm and application to structural optimization. Struct. Multidisc. Optim. 30, 219–226 (2005)
8. Fonseca, L.G., Barbosa, H.J.C., Lemonge, A.: A Genetic Algorithm with similarity-based fitness approximation for structural optimization. In: Proceedings of the XXVII Iberian Latin American Conference of Computational Methods in Engineering (2006)
9. Yoo, J., Hajela, P.: Immune Network Simulations in Multicriterion Design. In: Structural and Multidisciplinary Optimization, vol. 18, pp. 85–94. Springer, Heidelberg (1999)
10. Luh, G.C., Chueh, C.H.: Multi-objective Optimal Design of Truss Structure with Immune Algorithm. Computer and Structures 82, 829–844 (2004)
11. Hart, E., Timmis, J.: Application areas of AIS: The past, the present and the future. Applied Soft Computing 8, 191–201 (2008)

12. de Castro, L.N., Zuben, F.J.V.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems (2001)
13. de Castro, L.N., Timmis, J.: An Artificial Immune Network for Multimodal Function Optimization. In: CEC 2002, Proceedings of the 2002 Congress Evolutionary Computation, vol. 1, pp. 699–704 (2002)
14. Sompayrac, L.M.: How Immune System Works. Blackwell Publishing, Malden (2003)
15. Pina, A.A., de, A.C.H., Lima, B.S.L.P., Jacob, B.P.: Particle Swarm Optimization applied in design of risers structures Symposium of Computational Mechanics - SIMMEC (in Portuguese) (2008)

An Idiotypic Immune Network as a Short-Term Learning Architecture for Mobile Robots

Amanda Whitbrook, Uwe Aickelin, and Jonathan Garibaldi

School of Computer Science, University of Nottingham, UK, NG8 1BB
{amw,uxa,jmg}@cs.nott.ac.uk

Abstract. A combined Short-Term Learning (STL) and Long-Term Learning (LTL) approach to solving mobile robot navigation problems is presented and tested in both real and simulated environments. The LTL consists of rapid simulations that use a Genetic Algorithm to derive diverse sets of behaviours. These sets are then transferred to an idiotypic Artificial Immune System (AIS), which forms the STL phase, and the system is said to be seeded. The combined LTL-STL approach is compared with using STL only, and with using a hand-designed controller. In addition, the STL phase is tested when the idiotypic mechanism is turned off. The results provide substantial evidence that the best option is the seeded idiotypic system, i.e. the architecture that merges LTL with an idiotypic AIS for the STL. They also show that structurally different environments can be used for the two phases without compromising transferability.

1 Introduction

An important decision when designing effective controllers for mobile robots is how much *a priori* knowledge should be imparted to them. Should they attempt to learn all behaviours during the task, or should they begin with a set of pre-engineered actions? Both of these alternatives have considerable drawbacks; starting with no prior knowledge increases task time substantially because the robot has to undergo a learning period during which it is also at risk of damage. However, if it is solely reliant on designer-prescribed behaviours, it has no capacity for learning and adaptation.

The architecture described in this paper takes inspiration from the vertebrate immune system in order to attempt to overcome these problems. The immune system learns to recognize antigens over the lifetime of the individual (Short-Term Learning, STL), but also has knowledge of how to build successful antibodies from gene libraries that have evolved over the lifetime of the species (Long-Term Learning, LTL). This "two timescale" approach can be mimicked by coupling an idiotypic Artificial Immune System (AIS) scheme (STL phase) with a Genetic Algorithm (GA) that rapidly evolves sets of behaviours in simulation (LTL phase) to seed the AIS. This removes any need for hand-designing, permits more scope for creating adaptive solutions, and prevents robots from having to begin a task with no knowledge. The main focus here is describing the idiotypic AIS system (as the GA has already been treated in [1]), and testing whether the seeded system outperforms an unseeded one in both the real and simulated domains. In addition, the role of idiotypic selection in the STL is also examined by trialing systems that do not employ this feature.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 266 – 278, 2008.

The paper is arranged as follows. Section 2 discusses previous idiotypic AIS robot-controllers, and explains the potential benefits of coupling an LTL phase with an idiotypic system. Section 3 describes the test environments and problem used, and Section 4 presents a thorough description of the STL architecture. Section 5 highlights the experimental procedures used and Section 6 reports on and discusses the results obtained. Section 7 concludes the paper.

2 Background

The aim of this paper is chiefly to investigate whether there are distinct advantages to integrating LTL strategies (a GA run in fast simulation is used here) with STL strategies. In theory, the LTL phase should be able to provide the STL phase with unbiased (i.e. non-user-designed) starting behaviours, and the STL should permit the continued adaptation of the behaviours as the robot carries out its task in real time.

The STL phase used here is an idiotypic AIS network based on Farmer *et al.*'s [15] model of Jerne's [16] idiotypic network theory. In the model, antibody concentrations are dependent both on the antigens present, and on the other antibodies in the network, i.e. antibodies are suppressed and stimulated by each other as well as being stimulated by antigens. This means that the antibody that best matches the invading antigen is not necessarily selected for execution, which produces a more flexible and dynamic system. The theory has proved popular when designing AIS-based robot control systems, since it potentially allows great variability of robot behaviours (modelled by antibodies) in the face of changing environments (modelled by antigens).

However, past research has mostly been concerned with the structure and evolution of the antibody network, and little attention has been given towards the derivation and design of the antibodies themselves. For example, [3]–[7] all use GAs but evolve only the network links between the antibodies, which are hand-designed, fixed, and small in number. Reference [2] also uses a fixed set of pre-engineered antibodies. In contrast, the LTL phase of this research [1] uses a GA where six basic antibody-types are encoded with a set of six variable attributes that can take many different values, meaning that the system can evolve complete sets of simple but very diverse antibodies. These can then be passed to the STL phase, providing the potential to bestow much greater flexibility to the idiotypic system. In addition, the use of rapid simulations means that the AIS can be seeded within a very realistic time frame (less than twenty minutes) whereas most evolutionary work requires much longer to converge, sometimes even a number of days, which is prohibitive. For example, the systems developed in [8]–[11] have not overcome the unrealistic convergence-time problems.

The most important questions, however, are whether the evolved antibodies can be used effectively in an STL system, and whether such systems can cope with different environments, particularly the real world. Since environments can change, any form of STL needs to be adaptable as well as robust. Previous attempts at fusing STL and LTL include the use of neural networks, for example [12], which proves adaptable to different environments and across different platforms, but the system is trialed using a simple light-switching problem with no obstacles apart from the pen walls. In the experiments described here, more complex problems and much busier environments are

employed for testing. In [13] an evolutionary strategy is used for the STL phase. This provides continued adaptation, but deals with a maximum of only 21 behaviour parameters in the LTL phase. Here, behaviours are assembled in a piecewise fashion and from a huge pool of parameters, which should mean greater flexibility. In [14] the two learning phases are implemented simultaneously, but the system is trialed only in simple, structured environments. In addition, the authors claim an evolutionary period of only five minutes, but the results suggest that the robot was unable to avoid the obstacle prior to this. In contrast, the seeded STL system discussed here does not start until it has received the complete sets of GA-derived behaviours, so that it is fully ready to begin the task.

In order to establish that the initial seeding is extremely important in producing a robust and adaptable controller, unseeded systems (i.e. with no LTL phase) that begin with random behaviour sets are also tested. In addition, both the seeded and unseeded systems are run with and without the use of idiotypic effects, to establish the role of the idiotypic mechanism in providing flexibility. A hand-designed controller is also included to investigate how fixed strategies compare with variable ones. It uses a simple random wander to search for the target, a backward turning motion to escape collisions, and it steers the robot in the opposite direction of any detected obstacles. The research thus aims to investigate the following hypotheses:

- H_1 Seeded STL systems outperform unseeded STL systems.
- H_2 Seeded STL systems that employ idiotypic effects outperform seeded systems that do not.
- H_3 Seeded STL systems that employ idiotypic effects outperform fixed, hand-designed strategies.
- H_4 As long as the LTL-derived behaviours are sufficiently diverse, antibody replacement should not be necessary in the STL phase.

Reference [2] has already provided statistical evidence that idiotypic AIS systems are more effective than similar non-idiotypic ones, but it is restricted to a single robotic platform (Pioneer 3), the simulated domain, and only two different environments. This paper will hence also extend the research in [2] to include a different type of robot (e-puck), more environments, different problems, the real domain, an alternative RL strategy (see section 4.4), and a variable idiotope (see section 4.2).

3 Test Environments and Problem

The STL is conducted with an e-puck, a miniature mobile-robot with a small frontal camera and eight infra-red (IR) sensors that can detect the presence of objects up to a distance of about 0.1 m. Both virtual and real environments are used for testing. The simulated environments are worlds that have been designed using Webots [17] software, since the GA employs it, and many modules from the GA can be re-used for the AIS. Webots also permits easy transfer of control from the simulation to the real

robot. Two simulated worlds are considered, World 1 (see Fig. 1), and World 2 (see Fig. 2). In these the robot begins south of the central row of pillars and must detect and travel to the blue target-block in the north, avoiding collision with the obstacles, walls and pillars. In addition, a wandering e-puck acts as a dynamic obstacle. Once the robot has arrived at the target, the number of collisions c and the time to complete the task τ are recorded. The starting positions of the robots and target block are changed automatically after each run.

The real environment consists of a square wooden pen with sides 1.26 m long and 0.165 m high, (see Fig. 3). The mission robot must find and travel to the blue ball located in the pen, avoiding collisions. Once it has found the ball it must stop to signal that the target has been found. The obstacles, robots and ball are randomly placed in different starting positions after each run, so that the environment is slightly different in each case.

The seeded systems all take their starting antibody-sets from those created when the GA is run in the first world described in [1], i.e. a maze-world where the robot must track painted doors in order navigate to the end, (see Fig. 4). This world is employed in the LTL phase to show that the evolved behaviours do not have to be generated using the same environment and problem as in the STL phase.

Webots version 5.7.0 is used, running on GNU/Linux 2.6.9 (CentoOS distribution) with a Pentium 4 processor (clock speed 3.6 GHz). For both the real and virtual e-pucks the camera field-of-view is set to 0.3 radians, the pixel width and height to 15 and 3 pixels, and the speed unit for the wheels is set to 0.00683 radians/s.

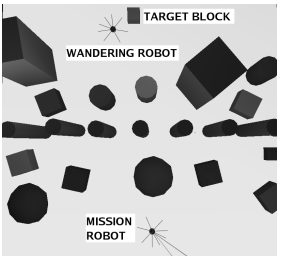


Fig. 1. Simulated World 1

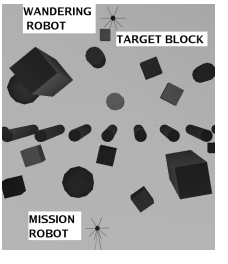


Fig. 2. Simulated World 2

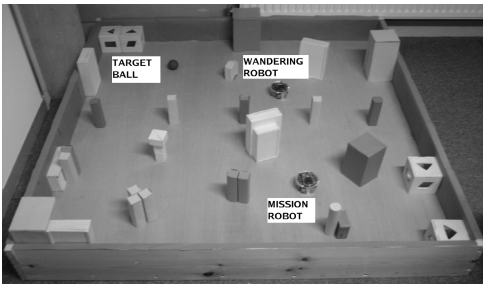


Fig. 3. Real World

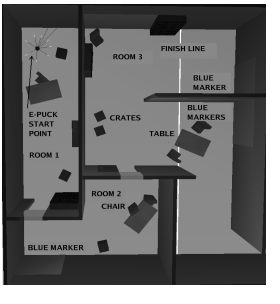


Fig. 4. GA Maze World

4 System Architecture

4.1 Antibodies and Antigens

There are eight antigens indexed 0-7, but only one presents itself at any instant. Either "0 - target unseen" or "1 - target seen" is active when no obstacles are present, (when the maximum IR reading V_m is less than 250). If V_m is between 250 and 2400 then either "2 - obstacle right", "3 - obstacle rear" or "4 - obstacle left" is active. If V_m is 2400 or more then "5 - collision right", "6 - collision rear", or "7 - collision left" presents itself.

There are six basic types of antibody, as listed in Table 1, and each possesses the attributes type T, speed S, frequency of turn F, turn angle A, direction of turn D, frequency of right turn R_f, angle of right turn R_a, and cumulative RL-score L. However, some types have null values for some attributes, and there are set limits to the values that the attributes can take.

Table 1. System Antibody Types

T	Description	*S* Speed Units / s		*F* % of time		*A* % reduction in speed of one wheel		*D* Either left or right		R_f % of time		R_a % reduction in right wheel-speed	
		MIN	MAX	MIN	MAX	MIN	MAX	1	2	MIN	MAX	MIN	MAX
0	Wander single	50	400	10	90	10	110	L	R	-	-	-	-
1	Wander both	50	400	10	90	10	110	-	-	10	90	10	110
2	Forward turn	50	400	-	-	20	200	L	R	-	-	-	-
3	Static turn	50	100	-	-	100	100	L	R	-	-	-	-
4	Reverse turn	300	400	-	-	20	200	L	R	-	-	-	-
5	Track markers	50	400	-	-	0	30	-	-	-	-	-	-

4.2 Creating the Paratope and Idiotope Matrices

An antibody set consists of eight behaviours, one for each antigen, and five distinct antibody sets are used. The 40 antibodies in the system can hence be represented as A_{ij}, $i = 0, \ldots, x\text{-}1$, $j = 0, \ldots, y\text{-}1$, where x is the number of sets and y is the number of antigens. For the seeded systems the evolved sets of antibody attribute values, their associated task completion times τ_i, and numbers of collisions c_i are read in from the file previously created when the GA was run. The STL system then calculates the relative fitness of each antibody set μ_i from:

$$\mu_i = \frac{1}{(\tau_i + \rho c_i)\sum_{k=0}^{x-1}(\tau_k + 8c_k)^{-1}}, \tag{1}$$

where $\rho = 8$ to give c equal weight compared to τ. It then produces a matrix of RL scores P_{ij}, which are analogous to antibody paratope values, as the scores represent a comparative estimate of how well each antibody matches its antigen, see [2]. The

elements of P_{ij} are calculated by multiplying the antibody's final RL score L_{ij} by the relative fitness μ_i of its set, and scaling approximately to between 0.00 and 1.00 using:

$$P_{ij} = \frac{L_{ij}\mu_i}{\varphi}. \tag{2}$$

Taking $\varphi = 20$ works here since the approximate maximum value $L_{ij}\mu_i$ can take is 20. For the unseeded systems the five antibody sets are generated at the start of the STL phase, by randomly choosing behaviour types and their attribute values. The initial elements of P_{ij} are also randomly generated, but always lie between 0.25 and 0.75 to try to limit any initial biasing of the selection.

For both seeded and unseeded systems, a matrix I_{ij} (analogous to a matrix of idiotope values, see [2]) is created by comparing the individual paratope matrix elements P_{ij} with the mean element value for each of the antigens σ_j. This is given by:

$$\sigma_j = \frac{\sum_{i=0}^{x-1} P_{ij}}{x}. \tag{3}$$

If P_{ij}, $i = 0, \ldots, x\text{-}1$ is less than σ_j, then an idiotope value I_{ij} of 1.0 is assigned, otherwise a value of zero is given. However, only one antibody in each set may have a non-zero idiotope. If more than one has a non-zero value, then one of them is selected at random and all the others are set back to zero. This avoids over-stimulation or over-suppression of antibodies.

The paratope matrix is adjusted after every iteration; first, because the active antibody's paratope value either increases or decreases, depending on the RL score awarded, and second, because all the paratope values are then re-calculated, so that the σ_j values are changed back to the initial mean values. The adjustment is given by:

$$P_{ij_{t+1}} = P_{ij_t} \frac{\sigma_{j_0}}{\sigma_{j_t}}, \tag{4}$$

where σ_{j0} represents the initial means and σ_{jt} represents the temporary means obtained after the active antibody has been scored. This adjustment helps to eliminate the problems that occur when useful antibodies end up with zero P_{ij} values. The idiotope is re-calculated, based on the latest P_{ij} values, after every 120 sensor readings, i.e. every 3.84 s, since the sensors are read every 32 milliseconds.

4.3 Antibody Selection Process

At the start of the STL phase each antibody has 1000 clones in the system, but the numbers fluctuate according to a variation of Farmer's equation:

$$N_{im_{(t+1)}} = bS_{im_{(t)}} + N_{im_{(t)}}(1 - k_3), \tag{5}$$

where N_{im} represents the number of clones of each antibody matching the invading antigen m. S_{im} is the current strength-of-match of each of these antibodies to m, b is a scaling constant and k_3 is the death rate constant, (see [2] for further details). The concentration C_{ij} of every antibody in the system consequently changes according to:

$$C_{ij} = \frac{\Phi N_{ij}}{\sum_{k=0}^{x-1}\sum_{l=0}^{y-1} N_{kl}}, \tag{6}$$

where Φ is another scaling factor that can be used to control the levels of inter-antibody stimulation and suppression (25 is used here).

The antibody selection process comprises three stages for idiotypic selection, but only one stage if idiotypic selection is not used. First, the sensors are read to determine the index of the presenting antigen m, and an appropriate antibody is selected from those available for that antigen. More specifically, the system chooses from antibodies A_{im}, $i = 0, \ldots, 4$, by examining the paratope values P_{im}. The antibody α with the highest of these paratope values is chosen as the first stage winner. If the index of the winning antibody set is denoted as n, then $\alpha = A_{nm}$. If idiotypic effects are not considered α carries out its action, and is assessed by RL, see section 4.4.

If an idiotypic system is used, then the stimulatory and suppressive effects of α on all the antibodies in the repertoire must be considered. As detailed in [2], this involves comparing the idiotope of α with the paratopes of the other antibodies to determine how much each is stimulated, and comparing the paratope of α with the idiotopes of the others to calculate how much each should be suppressed. Here, idiotypic selection is governed by equations (7)-(10), which are based on those in [2]. Equation (7) concerns the increase in strength-of-match value ε_{im} when stimulation occurs,

$$\varepsilon_{im} = k_1 \sum_{j=0}^{y-1} (1 - P_{ij})\, I_{nj}\, C_{ij}\, C_{nj}, \tag{7}$$

where k_1 is a constant that determines the magnitude of any stimulatory effects. The formula for the reduction in strength-of-match value δ_{im} when suppression occurs is:

$$\delta_{im} = k_2 \sum_{j=0}^{y-1} P_{nj}\; I_{ij}\, C_{ij}\, C_{nj}, \tag{8}$$

where k_2 governs the suppression magnitude. Hence, the strength-of-match after the second selection-stage $(S_{im})_2$ is given by:

$$(S_{im})_2 = (S_{im})_1 + \varepsilon_{im} - \delta_{im}, \tag{9}$$

where the initial strength-of-match $(S_{im})_1$ for each antibody is taken as the current P_{im} value. After the $(S_{im})_2$ values are calculated, the numbers of clones N_{im} are adjusted using (1) and all concentrations C_{ij} are re-evaluated using (2). The third stage calculates the activation λ of each antibody in the sub-set A_{im} from

$$\lambda_{im} = C_{im}(S_{im})_2. \tag{10}$$

The third-stage winning antibody β has the highest λ value. If p is the index of β's antibody set, then $\beta = A_{pm}$. When idiotypic selection is used, β carries out its action and it is β that is scored using RL rather than α, although α and β will be the same if $n = p$.

4.4 Reinforcement Learning and Antibody Replacement

Reinforcement learning scores the performance of an antibody by comparing old and new environmental information. Here, the antibody used in the previous iteration $A_{t\text{-}1}$ is assessed by examining the current and previous antigen codes m_t and $m_{t\text{-}1}$. Table 2 shows the RL score r awarded for each possible combination. The final score given is dependent on how many environmental changes have taken place, and whether the change is negative or positive, for example, moving away from an obstacle is a valuable improvement, and would yield a positive component of 0.1. The maximum cumulative-RL-score (or P_{ij} value) allowed is 1.00, and the minimum P_{ij} value is 0.00.

The P_{ij} values are also affected when the antigen code has remained at 0 for more than 250 iterations, as this means that the robot is spending too much time wandering and has not found anything. It is important to recognize this behaviour as negative, as otherwise robots may be circling around on the spot, never achieving anything, but receiving constant rewards. The non-idiotypic case reduces the cumulative-RL-score by 1.0, and the idiotypic case reduces it by 0.5, as pre-trials have shown that non-idiotypic robots require a more drastic change to break out of repeated behaviour cycles. The same P_{ij} adjustments are also made if there have been more than 15 consecutive obstacle encounters, as this may indicate that a robot is trapped.

Following RL, the paratope values are scaled using (4). In the case of the unseeded trials, replacement occurs for all antibodies with P_{ij} less than 0.1. When this takes place, a new antibody is created by randomly choosing a behaviour type and its attribute values. Antibody replacement is not used in the seeded systems, since H_4 is directly concerned with establishing whether this is necessary.

Table 2. Reinforcement Scores

Antigen code		***r* score**	**Reinforcement status**
Old	**New**		
0	0	0.05	Reward – No obstacles encountered
1	0	-0.10	Penalize - Lost sight of marker
2-7	0	0.10	Reward - Avoided obstacle
0	1	0.10	Reward - Found marker
1	1	0.00 to 0.05	Reward – Kept sight of marker (Score depends on orientation of marker with respect to robot)
2-7	1	0.20	Reward - Avoided obstacle and gained or kept sight of marker
0	2-7	-0.05	Penalize – Encountered obstacle
1	2-7	-0.05	Penalize – Encountered obstacle
2-7	2-7	-0.40 to 0.50	Reward or Penalize (Score depends on several factors)

5 Experimental Procedures

Before any of the seeded STL-phase tests take place, the GA is run once in the maze world, in accordance with the procedures described in [1], to obtain the initial seeding. Five independent populations of ten robots and a mutation rate of 5% are used, as recommended in [1]. Following this, 30 STL trials are performed in each of the two simulated worlds, and 20 are completed in the real world. This is done for each of the following systems; seeded with idiotypic effects, seeded with RL only, unseeded with idiotypic effects, unseeded with RL only, and a hand-designed controller. In the unseeded simulated-worlds two separate sets of experiments are conducted with two different initially-random behaviour sets R_1 and R_2. The real-world unseeded experiments use only R_1 since they have to run in real time and are hence much more time consuming to carry out.

In the idiotypic systems b is set to 100, k_3 is set to zero, and k_1 and k_2 are set at 0.85 and 1.10 respectively. These values are chosen in order to yield a mean idiotypic difference rate of approximately 20%, as this is advised in [2]. N. B. An idiotypic difference occurs when the antibodies α and β are different. For all experiments, the time taken τ and the number of collisions c are capped at 4000 s and 100 respectively. Any runs that exceed either of these limits are counted as failed runs. The fitness f is calculated as:

$$f = \frac{\rho c + \tau}{2}, \tag{11}$$

where $\rho = 8$ as before. A run finishes when the robot has detected three consecutive instances of more than 40 blue pixels in the ball image, so that it is "aware" of having found its target. Standard two-tailed t-tests are applied to compare the various systems, and differences are accepted as significant at the 99% level only.

6 Results

Table 3 shows the mean c, τ, and f values for each of the systems in each of the worlds, and Table 4 presents the significant difference levels when the systems are compared. Table 5 displays the failure rates, indicating the percentage of failures due to an excessive number of collisions, running out of time, and overall.

In all of the worlds, both simulated and real, the system with the lowest c, fastest τ, and best f is the seeded idiotypic system. When compared with the unseeded systems it is significantly better in all cases, i.e. for all of the metrics, in all the worlds, and irrespective of whether the unseeded systems use idiotypic effects, or which random behaviour set is used.

However, when the non-idiotypic seeded system is compared with the unseeded systems, although its performance is better in all cases, it is not always significantly better. Most of the significant differences arise when comparing seeded and unseeded systems that do not use idiotypic effects. In these cases, c is always significantly better for the seeded system, and, when R_2 is used in unseeded system, the seeded one is

always significantly better. When the unseeded system employs idiotypic effects and the seeded system does not, there is a marked drop in the percentage of significant differences, although many of the collision comparisons are significantly better for the seeded system.

When the seeded idiotypic system is compared with the seeded non-idiotypic system, the idiotypic system performs better in all cases, and significantly better in most cases. However, when the unseeded systems are compared in this way, although the idiotypic system consistently performs better, none of the differences are significant.

The seeded idiotypic system surpasses the hand-designed controller in all cases (except for a tie in c in Simulated World 2), and more than half of these differences are significant overall. However, in the real world all of the differences are significant. It appears that the hand-designed controller performs very well in the simulator in terms of c, but poorly for τ, whereas in the real world it performs badly for both of these metrics. The seeded idiotypic system works well in the real world and in the simulator for both c and τ. In fact, in the real world it proves significantly better than all of the other systems trialed, for all metrics.

Table 3. Mean c, τ, and f. (S = seeded, U = unseeded, IE = idiotypic effects, RL = reinforcement learning, HDC = hand-designed controller)

System	Set	Simulated World 1			Simulated World 2			Real World		
		c	τ	f	c	τ	f	c	τ	f
SIE	-	**1**	**562**	**284**	**2**	**659**	**336**	**5**	**283**	**161**
SRL	-	8	1298	679	4	1113	573	23	904	544
UIE	R_1	26	1513	862	26	1530	868	96	1384	1074
URL	R_1	45	2150	1253	35	1732	1006	100	1678	1239
UIE	R_2	20	1720	941	48	1578	981	-	-	-
URL	R_2	35	2214	1246	54	2137	1285	-	-	-
HDC	-	2	1362	688	**2**	1256	636	44	1439	897

Table 4. Significance Levels (S = seeded, U = unseeded, IE = idiotypic effects, RL = reinforcement learning, HDC = hand-designed controller)

Systems		Set	Simulated World 1			Simulated World 2			Real World		
			c	τ	f	c	τ	f	c	τ	f
SIE	**SRL**	-	**100**	**100**	**100**	**98**	**96**	**97**	**99**	**99**	**100**
SIE	**HDC**	-	85	**100**	**100**	33	**97**	**97**	**100**	**100**	**100**
SIE	**UIE**	R_1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
SIE	**URL**	R_1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
SIE	**UIE**	R_2	**99**	**100**	**100**	**100**	**100**	**100**	**-**	**-**	**-**
SIE	**URL**	R_2	**100**	**100**	**100**	**100**	**100**	**100**	**-**	**-**	**-**
SRL	**UIE**	R_1	**98**	49	72	**99**	83	92	**100**	85	**99**
SRL	**URL**	R_1	**100**	**99**	**100**	**100**	94	**98**	**100**	**96**	**100**
SRL	**UIE**	R_2	91	82	89	**100**	86	**98**	-	-	-
SRL	**URL**	R_2	**100**	**99**	**100**	**100**	**100**	**100**	-	-	-
UIE	**URL**	R_1	87	90	93	59	44	52	68	53	57
UIE	**URL**	R_2	82	81	87	40	86	84	-	-	-

Table 5. Percentage Failure Rates (S = seeded, U = unseeded, IE = idiotypic effects, RL = reinforcement learning, HDC = hand-designed controller)

System	Set	Simulated World 1 (%)			Simulated World 2 (%)			Real World (%)			Mean (%)		
		c	τ	Tot	c	τ	Tot	c	τ	Tot	c	τ	Tot
SIE	-	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SRL	-	**0**	3	3	**0**	7	7	10	5	10	3	5	7
UIE	R_1	23	17	30	20	13	23	95	10	95	46	13	49
URL	R_1	43	30	57	33	23	47	100	20	100	59	24	68
UIE	R_2	17	20	37	43	17	43	-	-	-	30	18	40
URL	R_2	30	30	47	50	27	53	-	-	-	40	28	50
HDC	-	**0**	20	20	**0**	17	17	10	25	35	3	21	24

Furthermore, the seeded idiotypic system is the only scheme that consistently displays a 0% failure rate. Failure rates are reasonably low (7% overall) for the non-idiotypic seeded system, but reach unacceptable proportions for the hand-designed controller (24% overall) and the idiotypic unseeded system (49% and 40% overall). The non-idiotypic unseeded system is clearly the worst option with overall fail rates of 68% and 50%. Moreover, the actual number of collisions for failing robots is often of the order of thousands for unseeded real-world systems, which renders the method entirely unsuitable.

These observations represent very strong statistical evidence in support of H_1 and H_3, i.e. they recommend the use of GA-seeded systems over both unseeded systems and fixed, user-designed systems. In particular, there is over-whelming statistical evidence in favour of using a seeded idiotypic system over any unseeded system, with all tests proving highly significant. In addition, the results provide some evidence to uphold H_2, since robot performance appears to be further enhanced by incorporating an idiotypic network into the STL architecture. In the seeded idiotypic system the GA provides immediate knowledge of how to begin the task, and the idiotypic AIS permits it to change and adapt its behaviour as the need arises. Without idiotypic effects, the seeded system has the same initial knowledge, but relies only on RL for adaptation, so it is less flexible. Although the hand-designed controller has built-in initial knowledge, it also proves inferior because of its inability to change the way it responds to an antigen. In contrast, the unseeded systems have no initial knowledge, and must acquire their abilities during the STL phase. This is a very slow process, even when idiotypic selection is used, because the search space is probably much too large given the time frame for completing the task. Moreover, the mechanism by which antibodies are replaced is not well developed; the robot is forced to select a random behaviour when it rejects an antibody, and could hence still be using random antibodies during the latter stages of task completion.

The results also demonstrate that behaviours derived in GA simulations can transfer extremely well to the real world, even when the simulated and real environments are very different. In addition, the tests show that the superiority of idiotypic AIS systems over RL-only systems (suggested in [2]) can be extended to the real world, other simulated worlds, and a different robotic platform. These experiments also uphold H_4, since the seeded idiotypic system exhibits a 0% fail rate in all cases, suggesting

that antibody replacement is not necessary when adequate seeding and a sufficiently adaptive strategy are in place.

7 Conclusions

This paper has described merging LTL (an accelerated GA run in simulation), with STL (an idiotypic AIS scheme), in order to seed the AIS with sets of very diverse behaviours that can work together to solve a mobile-robot target-finding problem. Results have shown that such seeded systems consistently perform significantly better than unseeded systems, and have also provided strong statistical evidence that the idiotypic selection process contributes towards this improved performance. The fusion of the two learning timescales has been shown to provide a rapid and realistic method for training robots in simulation, and an adaptable and robust system for carrying out real world activities.

References

1. Whitbrook, A.M., Aickelin, U., Garibaldi, J.M.: Genetic-Algorithm Seeding of Idiotypic Networks for Mobile-Robot Navigation. In: Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2008) (to appear, 2008)
2. Whitbrook, A.M., Aickelin, U., Garibaldi, J.M.: Idiotypic Immune Networks in Mobile Robot Control. IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics 37(6), 1581–1598 (2007)
3. Watanabe, Y., Ishiguro, A., Shirai, Y., Uchikawa, Y.: Emergent construction of behavior arbitration mechanism based on the immune system. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC), pp. 481–486 (1998)
4. Watanabe, Y., Kondo, T., Ishiguro, A., Shirai, Y., Uchikawa, Y.: Evolutionary construction of an immune network-based behavior arbitration mechanism for autonomous mobile robots. Electrical Engineering in Japan 123(3), 1–10 (1998)
5. Michelan, R., Von Zuben, F.J.: Decentralized control system for autonomous navigation based on an evolved artificial immune network. In: Proceedings of the 2002 Congress on Evolutionary Computation, vol. 2, pp. 102–1026 (2002)
6. Vargas, P.A., de Castro, L.N., Michelan, R.: An immune learning classifier network for autonomous navigation. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 69–80. Springer, Heidelberg (2003)
7. Krautmacher, M., Dilger, W.: AIS based robot navigation in a rescue scenario. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 106–118. Springer, Heidelberg (2004)
8. Floreano, D., Mondada, F.: Evolution of homing navigation in a real mobile robot. IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics 26(3), 396–407 (1996)
9. Marocco, D., Floreano, D.: Active vision and feature selection in evolutionary behavioural systems. In: From Animals to Animats: Proceedings of the 7th International Conference on Simulation of Adaptive Behaviour (SAB 2002), pp. 247–255 (2002)
10. Hornby, G., Takamura, S., Yokono, J., Hanagata, O., Yamamoto, T., Fujita, M.: Evolving robust gaits with AIBO. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 3040–3045 (2000)

11. Zykov, V., Bongard, J., Lipson, H.: Evolving dynamic gaits on a physical robot. In: Proceedings of The Genetic and Evolutionary Computation Conference (GECCO), Late Breaking Papers (2004)
12. Floreano, D., Urzelai, J.: Evolutionary robots with on-line self-organization and behavioural fitness. Neural Networks 13, 431–443 (2000)
13. Walker, J.H., Garrett, S.M., Wilson, M.S.: The balance between initial training and lifelong adaptation in evolving robot controllers. IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics 36(2), 423–432 (2006)
14. Keymeulen, D., Iwata, M., Kuniyoshi, Y., Higuchi, T.: Comparison between an off-line model-free and an on-line model-based evolution applied to a robotics navigation system using evolvable hardware. In: Artificial Life VI: Proceedings of the 6th International Conference on Artificial Life, pp. 199–209 (1998)
15. Farmer, J.D., Packard, N.H., Perelson, A.S.: The immune system, adaptation, and machine learning. Physica, D 2(1-3), 187–204 (1986)
16. Jerne, N.K.: Towards a network theory of the immune system. Ann. Immunol (Inst. Pasteur.) 125 C, 373–389 (1974)
17. Michel, O.: Cyberbotics Ltd – WebotsTM: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems 1(1), 39–42 (2004)

Conserved Self Pattern Recognition Algorithm

Senhua Yu and Dipankar Dasgupta

Department of Computer Science, University of Memphis, Memphis, TN 38152, USA
{syu,dasgupta}@memphis.edu

Abstract. Self-nonself model makes a lot of sense in the mechanisms of self versus nonself recognition in the immune system but it failed to explain a great number of findings. Some new immune theory is proposed to accommodate incompatible new findings, including Pattern Recognition Receptors (PRRs) Model and Danger Theory. Inspired from the PRRs model, a novel approach called Conserved Self Pattern Recognition Algorithm (CSPRA) is proposed in this paper. The algorithm is tested using the famous benchmark Fisher's Iris data. Preliminary results demonstrate that the new approach lowers the false positive and thus enhances the efficiency and reliability for anomaly detection without increase in complexity comparing to the classical Negative Selection Algorithm (NSA).

Keywords: Conserved Self Pattern Recognition Algorithm, Pattern Recognition Receptors Model, CSPRA, Artificial Immune System.

1 Introduction

Biologically inspired computing, in particular, the Artificial Immune Systems (AIS) is a promising solution to develop automated and adaptive defensive tools for current and future threats in the larger IT systems. Currently major types of AIS methods include Negative Selection Algorithms, Immune Network, Clonal Selection, and Danger Theory [1]. Negative Selection Algorithm (NSA) was first conceived mimicking the negative selection in the T-cell maturation process. Negative selection eliminates inappropriate and immature T-cells that bind to self antigens and thus allows the system to detect non-self antigens without mistakenly detecting self-antigens. NSA could be applied to one of the central challenges with computer security: determining normal and potential harmful activity, so it has attracted many AIS researchers and practitioners and found broader applications comparing to the other AIS models. Various features of the NSA make it by far the most popular algorithm, notably for anomaly detection [2]. However, NSA has its intrinsic limit by assuming foreign patterns intrusions and thus a high false positive error rate caused from this assumption [3]. On the contrast, non-self patterns would not necessarily indicate intrusions [4]. Within Biological Immune System (BIS), the traditional self-nonself model has failed to explain many findings. The new theories such as Modified Self-Nonself Model [5], Pattern Recognition Receptors (PRRs) Model [6-8], and Danger Model [5] are proposed. These models center on renewing the sense of self. A brief overview for these models will be provided in section 2.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 279–290, 2008.

Some efforts are made to tackle the above limit of NSA, for example, the more flexible boundaries are applied between self and non-self space using fuzzy rules [9]. Not all AIS are based on negative selection, the other major branch in AIS is emerged based on the new immune theory - Danger Theory.

BIS employs a multilevel defense against invaders through nonspecific (innate) and specific (adaptive) immunity. Adaptive immunity has the features of learning, adaptability, and memory, so many computer scientists and engineers use adaptive immunity as inspiration for producing immune-based defensive tools and algorithms. However, our interests have extended to the principle of innate immunity such as Pattern Recognition Receptors (PRRs) model. PRRs model assumes that Antigen Presenting Cells (APCs) are quiescent until they are activated via encoded PRRs that recognize conserved pathogen-associated molecular patterns (PAMPs) on bacteria [6]. The encoded PRRs allow APCs to discriminate between "infectious nonself" and "noninfectious self" [7].

Currently the AIS methods that are inspired from innate immunity are rare. The recent work [10] used the functionality of Dendritic Cell (DC) with Danger Theory as inspiration for developing the algorithm for anomaly detection. The author imported the concept of PAMPs in PRRs model to Danger Theory and regarded it as one of the danger signals within the tissue to determine the function that the DC will perform when the body is under attack. PAMPs are used as one of the signals in the experiment of detecting port scan attach and are represented as "unreachable destination" errors. The signal value is a normalized real-number recorded during data collection from the monitored system. That is, PAMPs in this application are regarded as signal molecules like the other signals in Danger Theory, which are established based on pre-defined anomalies. However, PAMPs in PRRs model are the conserved pattern within the antigen object and the signal refers to the process that PAMPs active the APCs rather than the PAMPs itself. In this paper, we report our on-going investigation – an immune based anomaly detection approach called Conserved Self Pattern Recognition Algorithm (CSPRA) from the abstraction of PRRs model. Our work extensively mimics the PRRs model but PAMPs are built based on normal behaviors. PAMPs in [10] are arbitrary, depending on application domain. In our algorithm, PAMPs are well-defined mandatory components in the training stage based on the inputted normal data in the system. The rest of the paper is organized as follows: Section 2 briefly introduces the immune theories on the mechanisms of self versus nonself recognition; Section 3 outlines the Pattern Recognition Receptors (PRRs) model; Section 4 describes the proposed algorithm; Section 5 reports the experimental results; finally, the conclusions are given in Section 6.

2 Self Versus Non-self

Of all the mysteries in modern science, the mechanisms of self versus nonself recognition in the immune system ranks at or near the top [11]. Self-nonself model suggested that the immune system functions by discriminating self (defined early in life) and nonself (anything that comes later), tolerating self and attacking nonself [5]. Although the self-nonself discrimination model has often served us well, it has failed to explain new findings. For example, why do organisms not attack their newly

changed tissues when they become older? Why do mammalian mothers not attack their newly lactating breasts that were not part of earlier "Self"? Why does the immune system not respond to vaccines composed of foreign proteins? [5]

To accommodate incompatible new findings, the concept of costimulation was added to self-nonself model, that is, the activation of individual effector cells requires not only ligation of the specific Receptors but also a second signal (name "costimulation") delivered from "stimulator" cell such as APCs. Costimulation principle suggested that the immune response is initiated by APCs. However, APCs are not antigen specific but they capture all sorts of self and foreign substances. Therefore, an unacceptable conclusion can be derived from cositmulation that the immunity cannot be directed against nonself.

In 1989, Janeway published the PRRs model. He proposed that APCs are quiescent until they are activated via a set of germ line - encoded PRRs that recognize conserved pathogen-associated molecular patterns (PAMPs) on bacteria [6]. We will describe this model in details in section 3.

Like self-nonself model, the PRRs model couldn't explain all of the findings in the immune system. For example, the conserved PAMPs are only experimentally discovered in *bacteria*, so this model couldn't explain why *viruses* stimulate immunity. The basis for discrimination in another immune theory - danger model was not centered around 'self' or 'non-self', but to the presence or absence of danger signals. It is thought that danger signals, such as those exposed to pathogens, toxins, mechanical damage, and so forth, are detected and processed through professional APCs. The danger model is presented in [5] and the applications of danger model to AIS were identified and discussed in depth in [12].

3 The Pattern Recognition Receptors (PRRs) Model

This paper takes the PRRs model and explores the relevance of this model to the application domain of computer security. We are going to give some background on this model in this section.

To solve the problems emerging from Self-nonself discrimination model and costimulation, as already stated in section 2, the basic idea for the PRRs model is that APCs can recognize evolutionarily distant pathogens. The self/nonself discrimination requests co-stimulation from APCs but APCs do not co-stimulate unless activated via encoded PRRs that recognize conserved pathogen-associated molecular patterns (PAMPs) on bacteria [6-7].

As shown in Fig.1, T cell or B cell would die if it recognized antigen (Signal 1) without the costimulation from APC (Signal 2). The nonclonal receptors in the surface of APC appear to detect common constituents of pathogenic microorganisms. These microbial structures cannot be modified by the pathogen without changing its core structure. Receptors for these structures have been selected over evolutionary time to provide broad-spectrum recognition of harmful foreign materials [7]. PRRs model only responds to the potential threat that is *very foreign* (evolutionally distant infectious non-self). The encoded PRRs allow APCs to discriminate between "infectious nonself" and "noninfectious self" [7], therefore, PRRs model is also known as the Infectious Nonself Model [5].

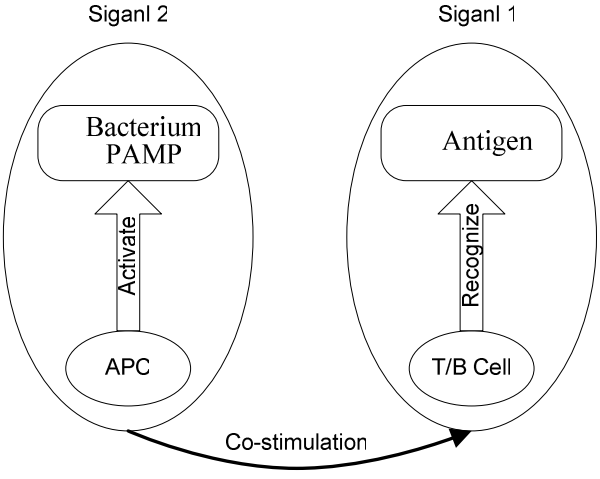


Fig. 1. Illustration of the Pattern-Recognition Receptors Model

Over the years, the PRRs model has been modified to account for some issues. Medzhitov et al discussed three strategies of immune recognition in the innate immune system of vertebrate animals, which are summarized as follows [8]:

- Recognition of "Microbial Nonself": the host recognizes conserved pathogen-associated molecular patterns that are unique to microorganisms and are not produced by the host. This strategy allows the innate immune system to discriminate between "infectious nonself" and "noninfectious self".
- Recognition of "Missing Self": this strategy relies on the detection of "marker of normal self". Such recognition is coupled with various inhibitory pathways that block initiation of immune responses against self. Markers of normal self are unique to the host and absent from microorganisms.
- Recognition of "Induced or Altered Self"**:** recognition of induced self is based on the detection of markers of abnormal self that are induced upon viral infection and cellular transformation. Markers of abnormal self tag the affected cells for elimination by the immune system.

4 CSPRA - Conserved Self Pattern Recognition Algorithm

NSA performs a high false positive error rate because of its assumption of foreign pattern. Inspired from relatively new immune theory – the PRRs model, the motivation of developing a new algorithm is to overcome the intrinsic limits of NSA – high false positive rate and improve the system performance. This section gives the general overview of the proposed algorithm. In our algorithm, both APCs Pattern Recognition and T cell Negative Selection are combined to detect anomalies in new samples, which efficiently reduces high false positive error rate.

4.1 Subspace Definition

PAMPs in the PRRs Model, as shown in Fig. 1, are pre-defined bacterial signatures, causing the maturation of APCs through the expression of mature cytokines [10]. Each high-dimension antigen (or self protein) in our system is regarded as the information vector (or object). Depending on application domain, we can explore a strong pattern of conservation, analogous to PAMPs in nature immune system, along a subset of their dimensions from the antigen objects. That is, there is a subset of the original dimensions in which the similarities are very high within the antigen objects. Such subset of dimensions is often referred to as a *subspace* in the area of data mining. The number of the dimensions in the subspace, as well as, the similarity of each dimension within the objects is varied for different application domain. The dimensions in the subspace are not necessary to be contiguous.

The system we proposed works over two distinct phases: Initialization and Detection. The first task for the system is to define the subspace based on the training dataset, which could be considered as preprocessing phase. In the intrusion detection system, it is practical to build up "Self" database from the normal behaviors of each process of interest or protected data but the abnormal behaviors usually remain unknown. We are interested in building an anomaly detection system, not an accurate simulation. Therefore, we change the concept of PAMP by considering it as conserved *self-associated* pattern rather than conserved *pathogen-associated* molecular patterns in nature immune system.

If the relationship between the objects and the dimensions of their feature space can be derived from the empirical data based on the scientists' lab results, the subspace, as described above, could be pre-defined. We are investigating the other approaches to establish the subspace.

4.2 Algorithm Overview

In the proposed algorithm, APC is treated as the detector capable of recognizing the conserved self pattern in the antigen objects. The representation and recognition of APC detectors are very different from the other AIS methods, which is described in section 5.2. In the system, T cell detectors are generated by using negative selection to eliminate invalid (self-recognized) candidates. Euclidean distance measure is employed to compute the affinity between T detectors and antigens and a partial matching rule is used, as described in [13].

PRRs model assumes that APCs are quiescent until they are activated via encoded pattern recognition receptors that recognize conserved pathogen-associated molecular patterns (PAMPs) [6]. To mirror this, T cell detectors in the proposed algorithm are first used to recognize the antigen according to negative selection. The co-stimulation of APC detector will not be conducted until the so-called *suspicious antigen* is encountered in the system. The suspicious antigen is defined as: 1) the affinity between the T cell detector and the suspicious antigen is very low; 2) the decision for abnormal (non-self) is made based on the other antigen epitope instead of the subspace where the conserved self pattern is located. For example, an antigen is defined as $m = < m_1, m_2 \ldots m_L >$, which can be considered as a point in an L-dimensional real-valued shape

space. If the antigen *m* epitope is <m_1, m_2> but the conserved pattern is <m_5, m_6> and the distance between the T detector *d* and the epitope <m_1, m_2> is greater than the defined threshold (low affinity), then the T cell detector *d* fails to recognize the antigen *m* without the co-stimulation of APC detector. The outline of the detection algorithm is illustrated in Fig. 2.

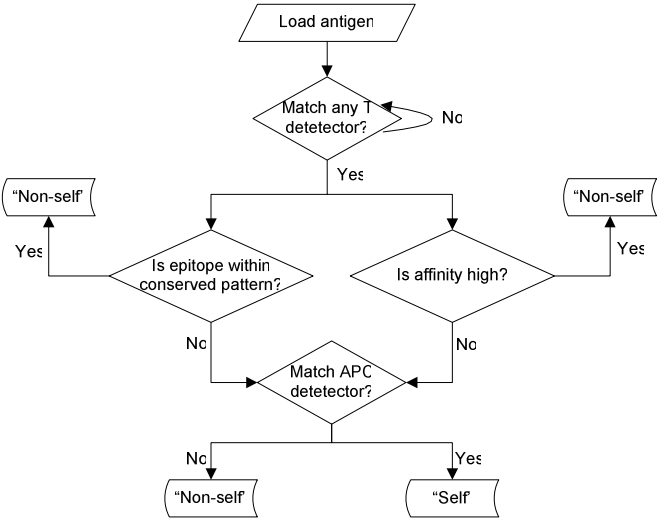


Fig. 2. Flowchart for Detection Algorithm

5 Empirical Evaluation

To study the algorithm property and evaluate the performance of our system, experiments were carried out with the famous benchmark Fisher's Iris Data. We start by describing some properties of Fisher's Iris Data, then go on to show that our algorithm can detect anomalies very efficiently comparing to NSA.

5.1 Some Properties of Iris Data

The Iris data set [14] was created by R. A. Fisher. This is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. There are four attributes in the data set. Adapted from the document in [14], the attribute information and some brief statistic analysis are presented in Table 1.

Table 1. The statistics of the attributes for Iris data set

Attribute Name	Min	Max	Mean	Class Correlation
Sepal Length	4.3	7.9	5.84	0.7826
Sepal Width	2.0	4.4	3.05	-0.4194
Petal Length	1.0	6.9	3.76	0.9490
Petal Width	0.1	2.5	1.20	0.9565

5.2 The Representation and Recognition of the APC Detectors

As stated before, APC detector represents for the conserved self pattern that are extracted from the collected "Self" data. In the algorithm, zero-based column (dimension) indexes are used to mark the positions of the conserved self pattern. For Iris data used in our experiments, as shown in Table 1, the class correlation for the attributes of both petal length and petal width is very high, therefore, the subset from both the 2nd and 3rd dimension of the original dimensions is considered as conserved self pattern. For the benchmark Iris Data, the APC detector is represented as *{<loc1, min, max, mean>, <loc2, min, max, mean>, …}*. For the training data of 50% Iris Setosa, we calculate the max, min and mean of all of the values in the 2nd and 3rd column and then the APC detectors are represented as follows:

{<2, 1, 1.9, 1.46>, <3, 0.1, 0.5, 0.248>}

During the detection phase, for each suspicious antigen to be co-stimulated by APC detector *d*, after extracting the conserved antigen peptide based on the position information in APC detector, for the above example, the conserved antigen peptide is projected from the attributes indexed as 2 and 3 in the original antigen, the distance between the antigen peptide *p* and the APC detector *d* is computed by

$$\text{Dist}(p, d) = \sum_{i=1}^{w} \frac{| pi - di |}{mi - ni} \tag{1}$$

where w is the number of the dimensions for the conserved pattern; m_i and n_i represent the lower and upper bounds of the i-th attribute in the entire training data; pi is the value of the i-th attribute for the antigen object to be examined; di is the mean of all of the values in the i-th column in the entire training data.

If the distance calculated by equation (1) is less than the pre-defined threshold, then the suspicious antigen peptide is similar to conserved self pattern and thus the suspicious antigen is concluded as "Self", possibly analogy to "Altered Self" in the PRRs model. Otherwise, the suspicious antigen is classified as "Non-self". As noted, APC recognition is occurred between each antigen peptide and the single APC detector, so the time complexity is *O(1)* for detecting each antigen. Comparing to NSA, the proposed algorithm acts as adding a special detector (APC detector) to the total size of the detectors but doesn't increase the complexity.

5.3 Algorithm Pseudo Code for Anomaly Detection

The algorithm implementation is very straightforward. The pseudo code is listed as follows:

```
//Initialization phase
GenerateTDetector()  //same as NSA
GenerateAPCDetector() //see section 5.2
```

```
//Detection phase
//System Input
S: set of antigen data
t1: threshold for T detector
t2: threshold for APC detector
t3: threshold for suspicious antigen
d: distance between T detector and the tested antigen
int loc[]: array for storing the positions for the antigen epitope

//Start recognition
for every si in S={si, i=1, 2, …}
    if(CheckWithTDetector(si, t1, d, loc ) = = true)  // d and loc will be returned
        if(d > t3) //low affinity
            if(loc doesn't match positions in APC) //not conserved pattern
                if(CheckWithAPCDetector(si, t2) = = true )
                    si is non-self
                else
                    si is self
            else
                si is non-self
        else
            si is non-self
    else
        si is self
end for

//implement CheckWithTDetector()
D: set of T detectors
w: sliding window size for projecting the antigen into peptides
P: set of peptides for tested antigen

bool CheckWithTDetector(si, t1, d, loc )
    ChopAgPeptide(si, w, P, loc) //like NSA, P and loc will be returned
        for every pi in P={pi, i=1, 2, …}
            for every dj in D={dj, j=1, 2, …}
                if(Dist(pi, dj) < t1) // compute Euclidean distance
                    d = Dist(pi, dj)
                    return true;
            end for
        end for
    return false;

//implement CheckWithAPCDetector()
d: APC detector
bool CheckWithAPCDetector(si, t2)
    if(Dist() < t2) //compute the distance with equation (1) in section 5.2
        return false;
    else
        return true;
```

5.4 Experimental Results

The results reported in this section are the summary of the 100 repeated tests for each method and parameters setting with the training data of 50% Iris Setosa. One of the three types of iris is treated as normal data, whereas the other two are considered as abnormal. For each experimental case, we take 50% of the normal data to train the system and the remaining 50% data are used to test the system capability of recognizing unknown normal data.

To be fair to the other approaches, it is very difficult to make reasonable comparisons between our method and some other AIS methods. But, it is meaningful and relatively fair to choose the classic NSA to compare with CSPRA. There are two reasons for this. Firstly, the PRRs model added additional layer of PAMPs to the Self-nonself model but kept the features of Self-nonself model. CSPRA naturally involves negative selection since it is inspired from the PRRs model, so we can employ the same values for the common parameters in the two methods to make the comparisons fair. Secondly, since CSPRA is targeted at overcoming the intrinsic limits of NSA, we must examine whether this goal is achieved. In this section, we report the experimental results and all conclusions are derived from such comparisons since it is noted that Iris data are smaller.

Since T detectors are randomly generated, different values for detection and false alarm rates are observed. Table 2 shows the mean and standard deviation of the results of 100 repeated experiments for CSPRA and NSA, respectively. The readers can reproduce the results with these parameters settings: sliding window size is 2; the threshold for T detector and APC detector are 0.1 and 1, respectively; T detector size is set to 35; the threshold for discriminating suspicious antigen is 0.02. These parameters are not meant to generate the best results from the system but enough to show that CSPRA outperforms the classical NSA. We found through the experiments that the threshold for T detector is the most sensitive parameter that influences the experimental results. Detection Rate (DR), False Alarm Rate (FA), and False Positive Rate (FP) are computed in equation (2) ~ (4). TP, FP, TN, and FN are defined in [13].

$$DR = TP/(TP+FN) \tag{2}$$

$$FA = FP/(FP+TN) \tag{3}$$

$$FP = FP/(FP+TP+FN+TN) \tag{4}$$

The results show that CSPRA has higher detection rate with far lower false alarm rate. As expected, Table 2 also shows that CSPRA has lower false positive rate, which is the major reason that CSPRA has higher performance.

Table 2. Experimental results for CSPRA and NSA

Training data	Algorithm	Detection Rate		False Alarm Rate		False Positive Rate	
		Mean	SD	Mean	SD	Mean	SD
Setosa	CSPRA	97.67	2.76	0.92	1.15	0.31	0.38
50%	NSA	97.24	3.19	4.76	2.54	1.59	0.85

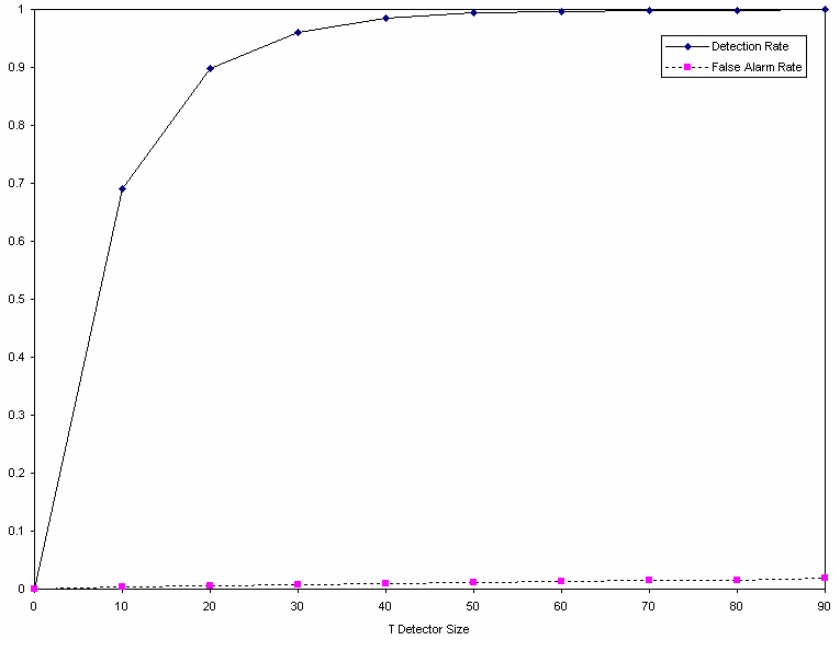


Fig. 3. T detector size's influence on Detection Rate and False Alarm Rate

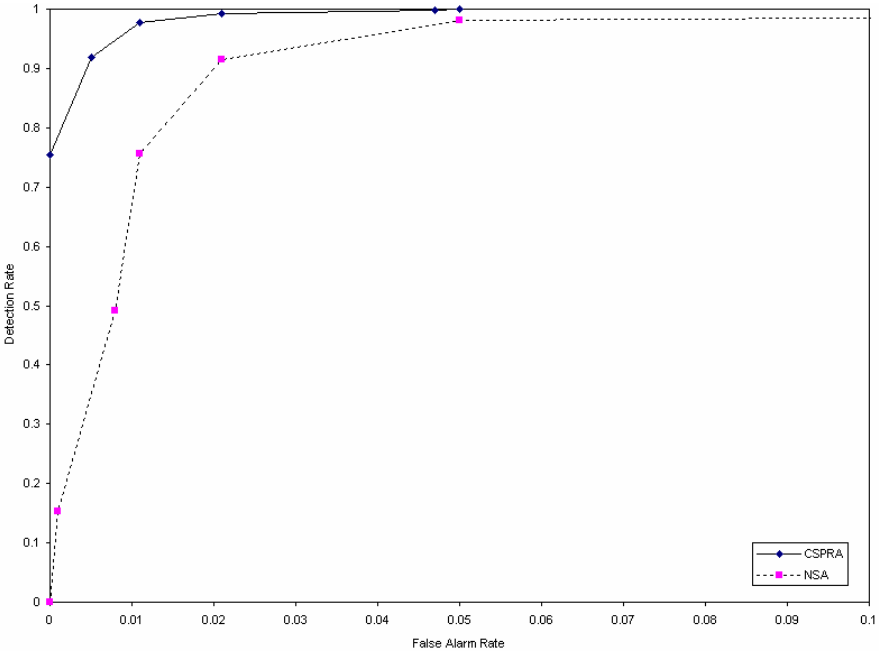


Fig. 4. ROC curves show the performance of CSPRA in comparison with NSA

The trend of T detector size's affect on the system performance is studied. As shown in Fig.3, the detector size initially influences the detection rate dramatically in CSPRA, however, the detection rate is close to 100% with lower false alarm rate when the detectors are increased to the size of around 40. V-detector is featured with smaller number of detectors and thus requires less space [15]. Our results are apparently comparable to V-detector. Fig. 3 also shows that false alarm rate is not sensitive to the detector size in CSPRA.

ROC (Receiver Operating Characteristics) curve reflects the tradeoff between false alarm rate and detection rate and is usually used to evaluate the system performance. Experiments are carried out to obtain different values for detection rate and false alarm rate by changing T detector thresholds. ROC curve is plotted for both NSA and CSPRA. As illustrated in Fig. 4, the performance of CSPRA is very promising. It always produces higher detection rate with the same false alarm rate as that in NSA.

6 Conclusions

In immunology, new theories are constantly being proposed explaining how the immune system responds to pathogenic material. This paper explores the relevance of the relatively new theory: PRRs Model in innate immunity to anomaly detection. By mimicking the concept of costimulation, PRRs and PAMPs in the PRRs Model, a novel immune-based approach called Conserved Self Pattern Recognition Algorithm (CSPRA) is presented. The algorithm is implemented and evaluated with the best known benchmark Fisher's Iris data. The experimental results show that the algorithm request smaller number of detectors and its performance is clearly better in comparison with the classical NSA, particularly for the discrimination between altered self and non-self. The major advantages of CSPRA are as follows:

- The pre-defined anomalies are required to establish the PAMPs in recent work [10]. However, only normal data are used in CSPRA during the training phase and the special APC detector gains the capability of recognizing the conserved self pattern in the antigen object from extracting the useful knowledge from the Self.
- False positive error rate for anomaly detection is greatly reduced with CSPRA, as the algorithm takes its inspiration from PRRs model, which allows APCs to discriminate between "infectious nonself" and "noninfectious self" and thus prevents from detecting altered self.
- The algorithm acts as adding only one robust detector to the total size of the detectors in NSA and thus doesn't increase the complexity of the system.
- It makes the system more efficient that APC detector is quiescent until the decision on "Non-self" made by T detectors is unsure.

Investigating the strategies for discovering the conserved pattern (self signatures) and finding real-world application are the main directions of our future work.

References

1. Dasgupta, D.: Advances in Artificial Immune System. IEEE computional Intelligence Magazine (2006)
2. Garrett, S.M.: How do we evaluate artificial immune systems? Evolutionary Computation 13(2), 145–178 (2005)
3. Aickelin, U., Greensmith, J., Twycross, J.: Immune System Approaches to Intrusion Detection – A Review. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 316–329. Springer, Heidelberg (2004)
4. Burgess, M.: Computer immunology. In: Proc. of the Systems Administration Conference (LISA 1998), pp. 283–297 (1998)
5. Matzinger, P.: The danger model: a renewed sense of self. Science 296(5566), 301–305 (2002)
6. Janeway Jr., C.A.: Approaching the asymptote? Evolution and revolution in immunology. In: Cold Spring Harbor Symp. Quant. Biol., vol. 54, pp. 1–13 (1989)
7. Janeway Jr., C.A.: The immune system evolved to discriminate infectious nonself from noninfectious self. Immunol. Today 13(1), 11–16 (1992)
8. Medzhitov, R., Janeway Jr., C.A.: Decoding the patterns of self and nonself by the innate immune system. Science 296(5566), 298–300 (2001)
9. Gomez, J., Gonzalez, F., Dasgupta, D.: An immuno-fuzzy approach to anomaly detection. In: proceedings of the 12th IEEE International Conference on Fuzzy Systems (FUZZIEEE), vol. 2, pp. 1219–1224 (2003)
10. Yeom, K.W.: Immune-inspired Algorithm for Anomaly Detection. In: Computational Intelligence in Information Assurance and Security. Studies in Computational Intelligence, vol. 57, pp. 129–154. Springer, Heidelberg (2007)
11. Koshland Jr., D.E.: Recognizing self from nonself. Science 248(4961), 1273 (1990)
12. Aickelin, U., Cayzer, S.: The danger theory and its application to artificial immune systems. In: proceedings of The First International Conference on Artificial Immune Systems (ICARIS 2002), pp. 141–148 (2002)
13. Dasgupta, D., Yu, S., Majumdar, N.S.: MILA - multilevel immune learning algorithm. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 183–194. Springer, Heidelberg (2003)
14. Iris Data Set, http://archive.ics.uci.edu/ml/datasets/Iris
15. Ji, Z., Dasgupta, D.: Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 287–298. Springer, Heidelberg (2004)

The Deterministic Dendritic Cell Algorithm

Julie Greensmith and Uwe Aickelin

Intelligent Modeling and Analysis,
School of Computer Science,
University of Nottingham, UK, NG8 1BB
{jqg,uxa}@cs.nott.ac.uk

Abstract. The Dendritic Cell Algorithm is an immune-inspired algorithm originally based on the function of natural dendritic cells. The original instantiation of the algorithm is a highly stochastic algorithm. While the performance of the algorithm is good when applied to large real-time datasets, it is difficult to analyse due to the number of random-based elements. In this paper a deterministic version of the algorithm is proposed, implemented and tested using a port scan dataset to provide a controllable system. This version consists of a controllable amount of parameters, which are experimented with in this paper. In addition the effects are examined of the use of time windows and variation on the number of cells, both which are shown to influence the algorithm. Finally a novel metric for the assessment of the algorithms output is introduced and proves to be a more sensitive metric than the metric used with the original Dendritic Cell Algorithm.

1 Introduction

Artificial Immune Systems (AISs) have developed significantly over the past five years, instigated by the creation of novel algorithms termed '2nd Generation AISs'. These AISs initially rely on interdisciplinary collaboration to use current research in immunology to produce algorithms which are both true to the underlying metaphor used as inspiration and perform well upon their resultant application domain. One such 2nd Generation AIS is the Dendritic Cell Algorithm (DCA), which is based on models of the dendritic cells (DCs) of the human immune system.

The original DCA was developed as part of the Danger Project [1], and formed the majority of Greensmith's thesis [3]. A prototype of the algorithm was first presented in 2005 [4] with a fully implemented real-time system version presented in 2006 [8]. The DCA has distinct advantages when applied to real-time computer security problems, as it has very low CPU processing requirements and does not require extensive training periods. All versions of the DCA to date have used a relatively large number of parameters and stochastic elements, such as random selection of cells and variable thresholds. Setting these parameters to the appropriate values has always been somewhat arbitrary, and thus has left the algorithm open to various criticisms. The use of various probabilistic elements was in part an artifact of the use of the Twycross' `libtissue` framework for the initial algorithm development. While this framework is useful for the rapid development of such AISs, one of the drawbacks for the DCA is the sheer amount of interacting entities.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 291–302, 2008.

As a result, it is still unclear which parts of the algorithm are responsible for its performance and for its time-dependent correlation properties. In order to push forward the DCA as a serious contender within biologically inspired computation, a thorough analysis of the algorithm itself must be performed: a task too complex when implemented within a large framework. Insight is needed into exactly what each component of the algorithm does and how detection is actually achieved. Despite avoiding a theoretical approach so far, the time has come to pick apart this algorithm and to break it down into a controllable deterministic system which is more accessible for the performance of various computational analyses and the various parameter relationships explored.

The aim of this paper is to describe, implement, and test a deterministic DCA (dDCA) to uncover its inner relationships and function. This paper is structured as follows, with background information present in section 2, section 3describing the dDCA and the new metric K_α. Experiments are described in section 4, with a discussion of results and conclusions presented in sections 5 and 6 respectively.

2 DCA Overview

Metaphorically, DCs are the crime-scene investigators of the human immune system, traversing the tissue for evidence of damage - namely signals, and for potential suspects responsible for the damage, namely antigen. More information regarding the function of natural DCs can be found in [10] with a distilled version for computer scientists presented in [3]. The DCA is derived from an abstract model of DC biology resulting in a population based algorithm which provides robust detection and correlation. Different cells process signals acquired over different time periods, generating individual 'snapshots' of input information which are correlated with antigens. The original DCA is described in detail in numerous sources including [7] and [3].

The majority of research performed with the DCA has been within the sphere of security. In particular, the works of Greensmith *et al.* have focussed on computer security applications. The algorithm to date has been successfully applied to port-scan detection [8] [6] [5], and upon comparison to a self organizing map performed well on the large dataset used, classifying 13 million antigens in under 100 seconds. In addition to her work, the DCA has also been applied to the detection of a novel threat on the internet, botnets [2], where the DCA produced high rates of true positives and low rates of false positives in comparison to a statistical technique. Outside of computer security Kim *et al.* have successfully applied the DCA to the detection of misbehaviour in wireless sensor networks, where again the algorithm showed a lot of promise. More recently in the work of Lay and Bate [9], the DCA is applied to the detection of overruns in the scheduling of processes, again with success.

The DCA is also showing promise in the area of robotic security as demonstrated by Oates *et al.* [11]. A proof of concept experiment is performed to demonstrate that the DCA could be used for basic object discrimination in a controlled environment. The same researchers have now extended this research into the theoretical domain [12] through frequency tuning analysis. This research has highlighted that the DCA exhibits filter properties and also suggests the importance of the lifespan limit. Their research

also contains two optimizations of the DCA which are used in this paper, namely a real valued representation of individual DC output and tissue centric processing of signals.

3 The Deterministic DCA (dDCA)

In this section the dDCA is formally described followed by a discussion of the modified features. In order to produce the deterministic version, it is necessary to make a number of assumptions and modifications to the original DCA:

- Both signals and antigen are required for the system to correctly function. If no signals are used, then the DCs will not exceed their lifespan limit and will not be able to present antigen. If no antigen are used, then the context has no subject.
- At minimum two signal categories are required, an activating signal and an inhibitory signal - the danger and safe signal respectively.
- A uniform distribution of lifespan values is used across the population. This allows for the study of the time-window effect in a repeatable and controllable manner.
- To provide reproducibility and for the ease of sensitivity a reduction in parameters is required from those used with the original DCA. As a result explicit antigen storage and sampling of the antigen population is removed, with all antigen data sampled by the DC population.
- Each DC in the population is exposed to identical input signal data and would process these signals in an identical manner. This results in the optimisation of the signal processing procedure, as the output signal values are calculated only once for the entire population, as suggested by Oates *et al.* [12].
- The output context value of an individual DC is reduced to one factor, $\bar{k}$, which negative numbers indicate a safe context and positive numbers indicating analogous to the previously used mature context. This is also derived from the theoretical analysis provided in Oates *et al.* [12].

One further modification is proposed for use with this system. This is the incorporation of an antigen profile. In previous implementations of the DCA, the string type antigens are stored in an 'antigen vector' data structure. This required the random selection of antigen by each DC and antigen overwriting. To ensure exact reproducibility the random sampling and storage is replaced by a simple array. In this array the value of the antigen is stored with the number of times a DC has collected antigens of this type. This reduces the required overhead as no dynamic memory management is required and leaves no concerns over denial of service due to the potential threat of antigen flooding.

Previous versions of the DCA featured in excess of 10 parameters, each of which were derived from empirical biological observation and through sensitivity analysis. The resultant algorithm contains three parameters. Firstly, the number of DCs must be defined - this is set to 100 as previously, but is experimented with in Section 4. Secondly the weighting schema for the signal processing. The signal processing equation used previously is modified for use with simplified weight values. As with the original DCA, the input signals are transformed to output signals. However a different procedure

is needed as the processing is performed in the tissue, the incorporation of k reduces the outputs from three to two and this is coupled with the reduction to two signal categories. The new signal processing procedure is shown in Equations 1 and 4, where S and D is the input value for the safe and danger signals respectively with 2 and 3 showing subsequent derivation thereof, c is the interim costimulation output signal and k is the interim context output value. Pseudocode for the implemented dDCA is given in Algorithm 1.

$$csm = S + D \tag{1}$$

$$k = (mature - semi) \tag{2}$$

$$k = (D - S) - S \tag{3}$$

$$k = D - 2S \tag{4}$$

input : Antigen and Signals
output: Antigen Types and cumulative k values

```
set number of cells;
initialise DCs();
while data do
    switch input do
        case antigen
            antigenCounter++;
            cell index = antigen counter modulus number of cells ;
            DC of cell index assigned antigen;
            update DC's antigen profile;
        end
        case signals
            calculate csm and k;
            for all DCs do
                DC.lifespan -= csm;
                DC.k += k;
                if DC.lifespan <= 0 then
                    log DC.k, number of antigen and cell iterations ;
                    reset DC();
                end
            end
        end
    end
end
for each antigen Type do
    calculate anomaly metrics;
end
```

Algorithm 1. Pseudocode of the deterministic DCA

3.1 Anomaly Metrics: MCAV and K_α

The mature context antigen value (MCAV) is calculated once all data is processed, derived from the output of the cells collected during run-time. This value is generated for each antigen type (α), where α is defined as a set of antigens of identical value. As the name suggests, the MCAV is a measure of the proportion of antigen presented by a fully mature cell as shown in Equation 5, where $MCAV_\alpha$ is the MCAV for antigen type α, M is the number of 'mature' antigen of type α, and Ag is the total amount of antigen presented for antigen type α.

$$MCAV_\alpha = \frac{M}{Ag} \tag{5}$$

This metric returns a value between zero and one, where the probability of an antigen type being anomalous increases as this value tends to one. This is a convenient, normalised output, to which an anomaly threshold can be applied. However, it fails to encapsulate the magnitude of the difference between positive and negative values of the presented $\bar{k}$. In the MCAV calculation a value of $\bar{k}$ of -1 is treated in exactly the same manner as a value of -200. The algorithm provides this information, hence it may be fruitful to incorporate this information into a more sophisticated metric.

K_α is implemented with the dDCA, and uses the magnitudes of the $\bar{k}$ values. This generates real valued anomaly scores and may assist in the polarisation of normal and anomalous processes. The process of calculating this anomaly score is shown in Equation 6, where k_m is the $\bar{k}$ value for DC_m, α_m is the number of antigen presented of type α by DC_m.

$$K_\alpha = \frac{\sum_m k_m}{\sum_m \alpha_m} \tag{6}$$

As this equation returns real valued numbers dependent on the actual values of the input signals used, we propose a method for defining an anomaly threshold, to allow for the classification of the antigen types analysed. This can be performed if the signals are known *a priori*. The number of signal instances and the equivalent processed total sum of the input signals. The threshold, T_K, is defined in Equation 7 with S_K, the weighted sum of all input signals, defined in Equation 8, where I_s is the number of pairs of signal instances, $\bar{i}$ is the mean number of iterations per cell incarnation, and D and S representing danger and safe signal values.

$$T_K = \frac{S_K}{I_s} * \bar{i} \tag{7}$$

$$S_K = \sum_{I_s} D - 2 \sum_{I_s} S \tag{8}$$

Once T_K is applied to the K_α values, antigen types with a value of over this threshold are classed as anomalous, and lower values classed as normal. If required, true and false positives can be derived from this information. A similar threshold can be derived from the MCAV, using the ratio of total danger signals to total safe signals present in the used dataset.

4 Experimental Analysis

4.1 Introduction

In this section initial tests are performed using the dDCA. This involves re-visiting a past dataset, namely the ping scan data used in Greensmith *et al.* [7] with one randomly selected set used to test the algorithm. In these experiments two aspects of the algorithm's function are examined:

- E0: A validation exercise to ensure the dDCA is correct.
- E1: The influence of variation in the number of cells.
- E2: Examination of 'time windows' and their effects on performance.

4.2 Testing Dataset

For these experiments one safe and one danger signal are used to provide the context information. As opposed to contriving artificial data, a dataset containing an outbound port scan is used. The object of using this data is that it is real-world data yet it is also relatively small, with approximately 25,000 antigens and 38 sets of danger and safe signal instances. The data is derived from a monitored remote shell session, where antigens are derived from process ID numbers and signals from monitored attributes of machine behaviour. Specifically, the danger signal is the rate of sending of outbound network packets, with the safe signal being the inverse rate of change of the packet sending rate. For more information of the necessity of these signals for port scan detection and for the mechanisms involve in port scanning please refer to [3].

In this dataset signals are updated once per second, with antigens generated as processes produce system calls. Both signals are normalised within a range of 0 to 50, based on maximum values derived in preliminary experiments. A graph of these signals is shown in Figure 1(a), where the mean danger signal value is 15.0 and mean safe signal value is 21.8. In terms of antigens, four processes of interest are captured by the antigen generator. These processes include two anomalous processes namely *nmap* the port scan process and *pts* a parent process of the nmap. Also included are two normal processes including *sshd* the remote shell facilitator process and *bash* the process of the actual monitored remote shell. The aim of the dDCA for these experiments is to produce high MCAV and K_α for the nmap and pts with lower values for the bash and sshd processes.

4.3 Experimental Setup

The deterministic DCA has two parameter values namely the number of cells and the lifespan limit. Unless specified otherwise, all experiments described use 100 artificial DCs with a maximum lifespan limit of 100 csm signal units. The increments of the lifespans are derived from the maximum limit divided by the number of cells. This is used to ensure an equivalent range of cells are present in each experiment. The T_K value used for this experiment is calculated as shown in Equation 9, where the number of signal instances is 38 and the mean number of iterations per cell incarnation is 2. The anomaly threshold for the MCAV is set to 0.69 based on the ratio of danger to

Table 1. MCAVs produced for dDCA versus Original DCA (mean of 3 runs)

Process ID	Original DCA	dDCA
nmap	0.999	0.969
pts	0.901	0.830
bash	0.711	0.623
sshd	0.070	0.202

safe signals within the dataset. The signal processing schema used is the one described previously in Equation 4. For the implementation, the dDCA is coded in C (gcc 4.0.1), with all experiments run on a 2.2 GHz MacBook Intel Core 2 Duo.

$$-57.4 = \frac{-1090}{38} * 2 \tag{9}$$

4.4 E0: Validation

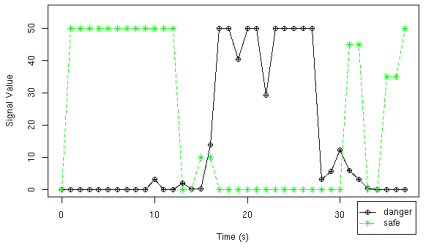
Before the dDCA can be used for these experiments, it must first be validated against the results generated by the original DCA. For this purpose, the results presented for the original DCA are derived from data used for Chapter 6 of [3]. The results of one run of the dDCA with default parameters are compared with three runs of the original DCA, with the MCAV results generated presented in Table 1. As shown in this table, the same trends are evident in both datasets. However, less polarisation between the normal and anomalous processes is shown with the dDCA. Despite such discrepancies, as similar trends are shown, we are confident that the dDCA is valid as a form of DCA.

4.5 E1: Cell Number Experiments

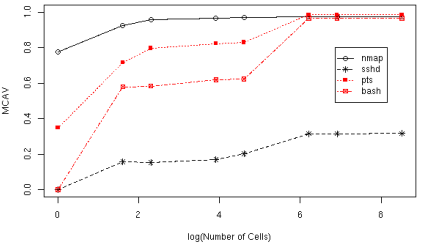
In this series, the number of cells used to process data are varied between runs. The set of cell numbers used is $n = \{1, 5, 10, 50, 100, 500, 1000, 5000\}$. Based on past sensitivity analyses of the cell numbers we expect the greatest variation between 1 and 100 cells. In addition to exploring this relationship, this experiment is used to generate statistics regarding the mean behaviour of the cell population. During these experiments, the number of antigen presented per cell per iteration, the number of iterations per lifespan and the number of cell resets are collated and mean values are calculated. Additionally, these experiments are timed to gain some insight into the scalability of the algorithm.

Both the MCAV and K_α values are shown for the four processes of interest for each cell number and we can use this information to assess the differences between the two output metrics. We predict that the real valued magnitude of K_α will produce more polarised results as it will provide discrimination between borderline cases and the more extreme, which of course is merely represented as 0 or 1 for the MCAV.

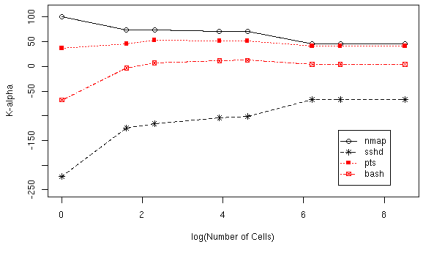
The results for the cell number experiments are shown in Figures 1(b) and 1(c). A graph of the timing results for the experiments are presented in Figure 1(d). Statistics regarding the cell behaviour information are presented in Table 2.



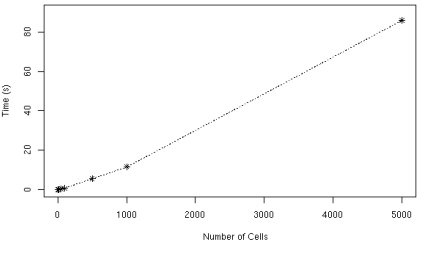
(a) Input Signals for the 38s session



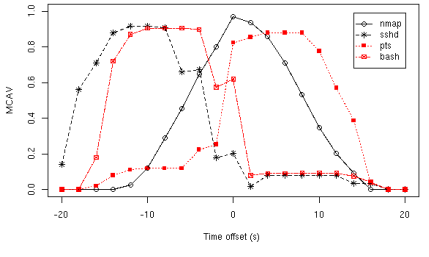
(b) MCAV of varying cell numbers



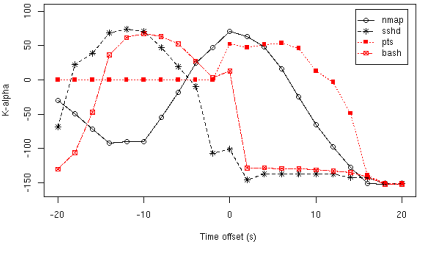
(c) K_α of varying cell numbers



(d) Execution Times of varying cell numbers



(e) MCAV for time-shifts



(f) K_α for time-shifts

Fig. 1. The input signal data is displayed in (a) with results for both series of experiments given in (b) to (f). Figures (b) and (c) show the MCAV and K_α values across a range of cell numbers plotted on a log-scale, (d) shows the execution times for varying the cell numbers, with (e) and (f) showing the MCAV and K_α with varying time delays.

4.6 E2: Time Window Experiments

It is assumed that the DCA performs correlation between antigen and signals based on time windows. These experiments are designed to ascertain if this is indeed the case. The nature of the time window effect created by the population of DCs is examined by shifting the position of the signals within the dataset. Each cell in the population has a lifespan, which defines the quantity of input signals the cell can process per incarnation. Having variable time windows should add robustness when the signals occur after the

Table 2. Cell behaviour statistics

Cell Number	Mean iterations	Mean incarnations
1	3.7	19.0
5	2.3	10.0
10	2.1	8.7
50	1.9	9.9
100	1.8	10.1
500	1.1	17.4
1000	1.1	17.4
5000	1.0	17.5

antigens, but we expect a reduction in DCA performance should the signals occur before the antigen.

While the cells create a type of moving average for the signals, this does not extend before the cell is initialised, and therefore signals appearing before antigen may result in a poor performance. A total of 20 extra datasets are created, with a maximum shift of 20 second for the signals before and after the original position, at two second increments. As with E1, both the MCAV and K_α values are calculated for each process of interest. These results are presented in Figures 1(e) and 1(f).

5 Experimental Analysis

5.1 E1: Cell Numbers

In E0 the dDCA is validated as fit for purpose. Subsequently when the number of cells is varied in E1 a noticeable effect on the performance of the DCA is indicated as shown in Figures 1(b) and 1(c). When the MCAV is used as the anomaly metric, an increase in the number of cells causes an increase in the MCAV for both pts and bash, though sshd and nmap do not increase to the same magnitude. The same trends are evident though less noticeable when using K_α for the bash and pts processes. This may be because it is difficult to assess if these processes, the parent processes of the nmap scan process, are actually anomalous or normal given that they have involvement in facilitating the scan itself. These two processes are borderline cases, and it appears that K_α provides improved information for this type of input data.

The sshd process which does not assist the scan has consistently low K_α values, well below the derived threshold of -57.4. It is interesting to note that as the number of cells used increases, the resultant output values converge. One possible explanation for this is that the lifespan limit is set incorrectly and maybe an improvement could be made if the range of these thresholds also increase in proportion to the number of cells.

Another explanation is that once the number of cells exceeds a certain limit, the capacity of the system exceeds the requirements of the input data, and therefore no matter how many extra cells are added, the resultant values remain similar. This is also shown in the summary statistics of the cell behaviour presented in Table 1. The results of the timed experiments are also encouraging, giving that the relationship between the number of cells and the execution time appears to be linear.

5.2 E2: Time Windows

The results of experiment E2 also show similar trends in comparison between MCAV and K_α, with the K_α values representing more precisely the classification of these processes. Therefore, K_α will be used in future for the assessment of our DCA experiments both empirical and theoretical. A marked difference is shown in particular for the nmap process between time offset -20 and zero and for the pts process also between -20 and zero.

Examination of the pts graphs show a moderately low MCAV value, yet when K_α is used, this value looks to remain stable at a level of 0. This could indicate that the pts process exhibits minor fluctuations around this point, with these fluctuations amplified by the binary classification of cells used in the MCAV, with K_α showing to be more sensitive to encapsulating such fluctuations.

In terms of the time window analysis two conclusions can be drawn from these graphs. Firstly, when the signals are delayed (time offset of 0 to 20), correct classification continues for almost 10 seconds, until the anomalous processes are classified as normal as they fall below T_K. Interestingly, improved results are shown with a delay of 2-4 seconds - which is equivalent to the average number of cell iterations per lifespan. Potentially the range of acceptable delay may be linked to a relationship between the number of iterations and the lifespan range itself, to which a formal analysis may be able to prove. Within the applications of the DCA in security so far, the signals are always updated after the antigens are generated, indicating one reason for why the DCA functions in the manner shown previously. These results suggest that the dDCA has the potential to be error tolerant to at least a five second lag in signal data, which is a desirable property for any behaviour based anomaly detection approach, as this reflects the situation often seen in real world intrusion data.

The opposite effect is shown when the signals are advanced ahead of the antigens. For the MCAV results both sets of processes, normal and anomalous, are classified incorrectly between time offset -20 and 0. A similar effect is seen for K_α for the same offset values. One explanation for this effect is that whilst cells produce a type of moving average, this is derived from information in only one direction i.e. the cells cannot incorporate information received before the start of their current incarnation. Therefore a reincarnated cell can only have knowledge of the signals which occur after its generation. While these results are interesting, a more formal analysis with contrived and controllable data must be performed in future in order to corroborate this tenet. This mirrors what is shown with natural DCs, as pathogenic infection (i.e. the presence of antigen) always occurs before the generation of danger signals.

6 Conclusions

In this paper a deterministic version of the DCA is proposed, implemented and tested. In addition to changes in the algorithm a new metric for the system's evaluation is proposed namely K_α which takes into account the magnitude of the output values produced by the DC population. The dDCA is compared to the original DCA using a port scan dataset used previously with the DCA. We are satisfied that while that results are not identical the values show similar trends, indicating that the essence of the DCA is

housed within the deterministic version. This version has several advantages, including the ability to replay experiments exactly, predictability of output and the reduction in the number of parameters required. All such factors have resulted in a version of the DCA which is simple to implement and can produce reliable, consistent results.

One of the remaining parameters of the dDCA is the number of cells used. As this number increases, discrimination between the processes is less obvious. While the cause of this effect still remains unclear it has given us insight into the limits of the system as it appears that there is a saturation point. For this particular dataset, this point is at 500 cells shown for both the MCAV and K_α. The metric K_α is tested for the first time in this experiment and is shown to be more sensitive to the minor fluctuations in the resulting output of the cells and provides a more precise overview of the classification of the various antigen types. To assess the implications of K_α, this metric should be applied to a wider range of problems.

Finally, timing discrepancies between signals and antigen are performed. As a result it is shown that should there be a delay for the input signals, within a tolerance range the dDCA can cope well with this delay. A potential relationship between the lifespan maximum limit and the number of iterations per cell incarnation may exist, though a more formal analysis is required to verify this effect. Conversely, if the signal data is advanced, severe misclassifications can occur, hence suggesting that the dDCA should not be applied to data where there is the potential for delayed antigen as performance may be impaired.

As future work we intend to further explore this new instantiation of the DCA. This investigation will involve a more in-depth study of the inherent relationships present within the algorithm in addition to extensive testing both on a range of real-world and synthetic data, and in comparison with other standard techniques such as support vector machines. This has the aim of selecting such parameters appropriately no matter what the application. In conclusion, the dDCA is a comparable and controllable form of the DCA and is a powerful tool necessary to further the understanding of this interesting immune-inspired algorithm.

Acknowledgements

This research is supported by the EPSRC (EP/D071976/1). Code optimisations courtesy of Gianni Tedesco.

References

1. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger theory: The link between AIS and IDS. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
2. Al-Hammadi, Y., Aickelin, U., Greensmith, J.: DCA for detecting bots. In: Proc. of the Congress on Evolutionary Computation (CEC), page tba (to appear, 2008)
3. Greensmith, J.: The Dendritic Cell Algorithm. PhD thesis, School of Computer Science, University Of Nottingham (2007)

4. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing Dendritic Cells as a novel immune-inspired algorithm for anomaly detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
5. Greensmith, J., Aickelin, U., Feyereisl, J.: The DCA-SOMe comparison: A comparative study between two biologically-inspired algorithms. Evolutionary Intelligence: Special Issue on Artificial Immune Systems (accepted for publication, 2008)
6. Greensmith, J., Aickelin, U., Tedesco, G.: Information fusion for anomaly detection with the DCA. Information Fusion (in print) (2008)
7. Greensmith, J., Aickelin, U., Twycross, J.: Articulation and clarification of the Dendritic Cell Algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 404–417. Springer, Heidelberg (2006)
8. Greensmith, J., Twycross, J., Aickelin, U.: Dendritic cells for anomaly detection. In: Proc. of the Congress on Evolutionary Computation (CEC), pp. 664–671 (2006)
9. Lay, N., Bate, I.: Improving the reliability of real-time embedded systems using innate immune techniques. Evolutionary Intelligence: Special Issue on Artificial Immune Systems (2008)
10. Lutz, M., Schuler, G.: Immature, semi-mature and fully mature dendritic cells: which signals induce tolerance or immunity? Trends in Immunology 23(9), 991–1045 (2002)
11. Oates, R., Greensmith, J., Aickelin, U., Garibaldi, J., Kendall, G.: The application of a dendritic cell algorithm to a robotic classifier. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 204–215. Springer, Heidelberg (2007)
12. Oates, R., Kendall, G., Garibaldi, J.: and. Frequency analysis for dendritic cell population tuning: Decimating the dendritic cell. Evolutionary Intelligence: Special Issue on Artificial Immune Systems (2008)

Artificial Immune Systems and Kernel Methods

T.S. Guzella[1,2], T.A. Mota-Santos[2], and W.M. Caminhas[1]

[1] Dept. of Electrical Engineering, Federal University of Minas Gerais,
Belo Horizonte (MG) 31270-010, Brazil
{tguzella,caminhas}@cpdee.ufmg.br
[2] Dept. of Biochemistry and Immunology, Federal University of Minas Gerais,
Belo Horizonte (MG) 31270-010, Brazil
tomaz@icb.ufmg.br

Abstract. In this paper, we focus on the potential for applying Kernel Methods into Artificial Immune Systems. This is based on the fact that the commonly employed "affinity functions" can usually be replaced by kernel functions, leading to algorithms operating in the feature space. A discussion of this applicability in negative/positive selection algorithms, the dendritic cell algorithm and immune network algorithms is conducted. As a practical application, we modify the aiNet (Artificial Immune Network) algorithm to use a kernel function, and analyze its compression quality using synthetic datasets. It is concluded that the use of properly adjusted kernel functions can improve the compression quality of the algorithm. Furthermore, we briefly discuss some of the future implications of using kernel functions in immune-inspired algorithms.

Keywords: Artificial Immune System, Affinity Functions, Kernel Methods, Immune Network, aiNet.

1 Introduction

Recently, Artificial Immune Systems (AISs), have emerged as a novel soft computing paradigm [1], seeking inspiration in the immune system for the development of computational models for solving problems. Most algorithms employ the concept of a so called "affinity function", which describes the degree of matching between two entities (a cell or antibody and an antigen). Usually, these affinity functions are obtained by adapting corresponding distance functions, so that the affinity between two entities is inversely proportional to their distance in some metric space, and the algorithms can be described in terms of distances.

In the framework for designing AISs proposed in [1], the design of the affinity function(s) follows the definition of the representation used for cells and molecules. In the case of systems employing real-valued representations, the Euclidean distance is one of the most commonly used affinity measures. This is inspired by early theoretical work by Perelson and Oster [2], which proposed the concept of a shape space, a metric space for quantifying the chemical interactions between molecules, where the Euclidean distance was originally used. However, in accordance with the convention followed in the machine learning community,

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 303–315, 2008.

we will use the term input space when referring to the shape space. This use of general affinity functions, without taking into consideration the characteristics of the target problem, has been recently criticized by Freitas and Timmis [3], who highlight the need to follow a problem-oriented approach in designing an AIS, in which the adoption of a certain affinity function is justified by characteristics of the target application.

The impact of using some affinity functions has been recently studied by some researchers. Hart [4] has shown the effects of the affinity function in idiotypic networks based on real-valued representations, influencing the size and dynamics of the resultant networks, pointing out the importance of carefully defining the affinity function and network parameters when applying a network to solve a problem. Recent work by Hart et al. [5] provides additional evidence of effects on the topology of the network, influencing its properties. In the context of negative selection algorithms, Stibor et al. [6] have conducted an in depth analysis of the use of the Euclidean distance, showing that coverage problems arise when dealing with high-dimensional data.

In parallel, kernel-based learning algorithms have been gaining an increasing focus in research, such as Support Vector Machines (SVMs) [7] and kernel PCA [8]. Kernel methods are based on mapping an input data point into a suitable Hilbert space, termed the feature space, allowing for very general representations of characteristics of the data being analyzed, and then performing computations in this new space. The underlying theory allows the manipulation of data in the potentially infinite-dimensional feature space without explicitly knowing the map from the input to the feature space.

In considering that several AISs can be seen as similarity-based algorithms, due to the use of distance functions, we analyze the application of kernel methods in immune-inspired models, discussing, in an informal way, how some algorithms can be modified to work in the feature space. Argued by Timmis [9] as an important line of investigation to allow the advancement of AIS, theoretical aspects have been receiving an increasing interest (see review in [10]). Therefore, grounding the use of affinity functions into a proper theoretical framework is an important step towards the design of new algorithms.

This paper is organized in the following way: section 2 presents a brief overview of the theory of Reproducing Kernel Hilbert Spaces, which provides a theoretical basis for the application of kernel functions. In sequence, section 3 discusses the applicability of kernel methods in some immune-inspired algorithms. As a practical application, section 4 derives and evaluates a kernel-based version of aiNet (Artificial Immune Network) [11]. Finally, section 5 presents the final conclusions of this work, along with future research directions.

2 Theory of Reproducing Kernel Hilbert Spaces

This section presents a brief discussion of the basic concepts of the theory of Reproducing Kernel Hilbert Spaces (RKHSs), closely following [8]. Throughout the discussion, we assume that the input space $\mathcal{X}$ is a non-empty set, and restrict

the presentation to real-valued kernels. In addition, the dot product between two vectors x and y is represented by $\langle x, y\rangle$. The dot product is the starting point in the theory of kernel methods due to the fact that it allows the generalization of several geometrical operations (such as projection, distances and the angle between two vectors). In learning algorithms, it is used to derive a notion of similarity between two elements (not necessarily vectors).

Definition 1. (Positive Definite (PD) Kernel) *A kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \Re$ is a PD kernel if, for any $X = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$, the $n \times n$ matrix with elements $k_{i,j} = k(x_i, x_j)$ is positive definite.*

In particular, for a PD kernel, $k(x, x) \geq 0$, $\forall x \in \mathcal{X}$, $k(x, y) = k(y, x)$ and the Cauchy-Schwarz inequality $|k(x, y)|^2 \leq k(x, x)k(y, y)$ holds (from which the triangle inequality can be derived). Defining a map $\phi : \mathcal{X} \rightarrow \mathcal{H}$, where $\mathcal{H}$ is the space of functions mapping X into $\Re$ (usually referred to as the feature space), $\phi(x)$ can then be seen as a function that assigns the value $k(x, y)$ to $y \in \mathcal{X}$, thereby transforming each point x into a function. Given an arbitrary set $X = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$, considering the vector space defined by linear combinations $f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$, the dot product can be computed as:

$$\langle f, f\rangle = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \tag{1}$$

where the non-negativity follows from the fact that the kernel k is PD, implying that $\langle \cdot, \cdot \rangle$ in the feature space is also a PD kernel. The concept of the space where the mapped patterns $\phi(x)$ lie is formalized through the definition of a Reproducing Kernel Hilbert Space.

Definition 2. (Reproducing Kernel Hilbert Space (RKHS)) *A Hilbert Space of Functions $\mathcal{H}$ is a RKHS with kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \Re$, possessing a dot product $\langle \cdot, \cdot \rangle$ and a corresponding norm $\|f\| = \sqrt{\langle f, f\rangle}$ if the following two conditions are satisfied:*

1. *k has the reproducing property:*

$$\langle k(x, \cdot), f\rangle = f(x) \ \forall f \in \mathcal{X} \tag{2}$$

and, in particular:

$$\langle k(x, \cdot), k(y, \cdot)\rangle = k(x, y) \tag{3}$$

2. *k spans $\mathcal{H}$, or, in other words, $k(x, \cdot)$, as a function of $x \in \mathcal{X}$, belongs to $\mathcal{H}$*

Due to the requirement of positive definiteness, the denomination kernel is usually used when referring to PD kernels, a convention followed by this paper from this point on. Using the reproducing property shown in equation 3, it can be verified that:

$$\langle \phi(x), \phi(y)\rangle = k(x, y) \tag{4}$$

and applying the kernel function to points x and y is equivalent to calculating the dot product between the mapped points $\phi(x)$ and $\phi(y)$. Therefore, kernel

functions allow the calculation of dot products in the resultant RKHS, without explicitly knowing the map ϕ. In addition, it is possible to calculate the distance between two mapped points, through the relation:

$$\|\phi(x) - \phi(y)\|^2 = k(x,x) + k(y,y) - 2k(x,y) \geq 0 \tag{5}$$

which is non-negative due to the fact that k is PD. However, it should be noted that there is a larger class of kernels than can be used to determine the distance in the feature space (referred to as conditionally positive definite kernels, [8]), although requiring some additional adaptations.

Finally, we present some examples of commonly used PD kernels, such as Gaussian:

$$k(x,y) = \exp\left(-\frac{1}{2\sigma^2}\|x-y\|^2\right) \tag{6}$$

polynomial:

$$k(x,y) = (\langle x,y\rangle + c)^d \tag{7}$$

and the inverse multiquadric kernels:

$$k(x,y) = \left(\sqrt{\|x-y\|^2 + \sigma^2}\right)^{-1} \tag{8}$$

The parameters $\sigma > 0$, $c \geq 0$ and $d \in \mathbb{N}$ determine the shape of the mapped points in the feature space, and their appropriate adjustment is crucial for a good performance of the algorithms employing such kernel functions. In addition, due to the fact that the choice of a kernel for an application is rather arbitrary, there is an increasing focus on the development of kernel functions incorporating prior-knowledge (e.g. [8]).

3 Applicability of Kernel Functions in Immune-Inspired Algorithms

In this section, we briefly discuss the applicability of the theory of RKHS in various immune-inspired algorithms. A potential advantage of using kernel functions is that these allow for more general representations of data dependencies, which can improve the performance of some algorithms. In addition, from the discussion presented in the previous section, it follows that the only theoretical requirement is that $\mathcal{X}$ is a non-empty set. As AIS are not restricted to real-valued representations (see [3] for a discussion of the representations used in some models), the framework of kernel methods fits, initially, nicely in this area. In the following paragraphs, we center the discussion in three families of algorithms: positive/negative selection, the dendritic cell algorithm and immune network approaches, focusing on real-valued representations.

3.1 Positive and Negative Selection

In positive/negative selection approaches (e.g. [12]), a set D, containing detectors, is checked against a test point x to determine if it is indicative of normal or

anomalous behavior (usually referred to as self and non-self, respectively). This procedure can be described for both algorithms by equation 9:

$$f(x) = \theta\left(\sum_{s_i \in D} \theta\left(b_i - d\left(x, s_i\right)\right)\right) \tag{9}$$

where $\theta(\cdot)$ is the step function defined by $\theta(x) = 1$, if $x > 0$ or 0, otherwise, and b_i is the activation threshold of the i-th detector. The expression $\theta\left(b_i - d\left(x, s_i\right)\right)$ represents the activation of the i-th detector, which happens if its distance for the test point x is smaller than the threshold b_i. In positive detection schemes, it follows that, if $f(x) = 1$, then x is classified as normal, while, in negative detection algorithms, $f(x) = 1$ indicates that x is anomalous (non-self).

An analysis of equation 9 indicates that replacing the commonly used Euclidean distance in positive/negative selection algorithms with a kernel function, so that the distance is evaluated in feature space (i.e. equation 5) should have a minor impact in the performance of such algorithms. Due to the fact that the activation of one detector does not influence the remaining detectors, the evaluation of the distance in feature space merely alters the recognition region of each detector in the input space (defined by $S_i = \{x : d(x, s_i) \leq b_i, x \in \mathcal{X}\}$). As an example, using the Gaussian kernel (equation 6) to calculate the distance can be seen as merely changing the radius of detection in comparison with the one obtained with the Euclidean distance.

3.2 Dendritic Cell Algorithm

The Dendritic Cell Algorithm (DCA) [13] is a recent proposal in the area of AIS. It is based on the behavior of dendritic cells sampling antigens and signals from the environment, and assuming a migration behavior depending on the sampled signals. Given four input signals (danger, PAMP, safe and an inflammatory signal), three output signals are derived, indicating the co-stimulation of each dendritic cell, along with a mature (a pro-inflammatory phenotype) and a semi-mature output signal (anti-inflammatory phenotype). In the case that the inflammatory signal is constant, the output signals can are given as a linear combination of the input signals:

$$\Psi_{cs} = w_d^{cs} I_d + w_p^{cs} I_p + w_s^{cs} I_s \tag{10}$$

$$\Psi_{mt} = w_d^{mt} I_d + w_p^{mt} I_p + w_s^{mt} I_s \tag{11}$$

$$\Psi_{sm} = w_d^{sm} I_d + w_p^{sm} I_p + w_s^{sm} I_s \tag{12}$$

where Ψ_{cs}, Ψ_{mt} and Ψ_{sm} are the co-stimulation, mature and semi-mature output signals, I_d, I_p and I_s are the danger, PAMP and safe input signals, and the w's are constants. It can be seen that, in this setting, the output signals can

be represented as dot products[1] between a vector containing the input signals ($I = \left[I_d \; I_p \; I_s \right]^T$) and another vector containing the appropriate constants:

$$\Psi_{cs} = \langle w^{cs}, I \rangle \tag{13}$$

$$\Psi_{mt} = \langle w^{mt}, I \rangle \tag{14}$$

$$\Psi_{sm} = \langle w^{sm}, I \rangle \tag{15}$$

In this case, it follows that each output signal is obtained by multiplying the length of the appropriate weight vector by the projection of I onto the weight vector ($\|I\| \cos(\alpha)$, where α is the angle between I and the w vector).

Therefore, even though the DCA does not employ affinity functions, kernel functions could be applied to it, replacing the dot products in equations 13-15. However, the meaning of such modification is not clear at present, due to the fact that the relevant parameters of the algorithm have been derived from experimental data. It should become clearer as the general mathematical properties of the algorithm are investigated.

3.3 Idiotypic Network Algorithms

Idiotypic network algorithms (also called immune network algorithms) are based on a network theory of the immune system. Two examples are aiNet, proposed by de Castro and Von Zuben [11], and the network-based AIS presented by Timmis et al. [14]. The use of kernel functions in these algorithms should have a noticeable impact on these algorithms, due to the fact that the affinity between antibodies or B cells and antigens usually affects the structure of the networks (e.g. [4]). In particular, in the next section, we consider, as a practical example of the incorporation of kernel functions in AIS, the derivation of a modified aiNet algorithm, which operates in the feature space, and analyze how its performance is influenced.

4 A Practical Application: aiNet

4.1 Derivation of Kernel-Based Version of aiNet

In this section, we present a kernel-based version of aiNet, an immune network algorithm proposed in [11]. This algorithm was chosen due to the recent work of Stibor and Timmis [15], which investigates its compression quality, using the originally proposed Euclidean distance as affinity function. In that work, it was verified that aiNet may face problems when dealing with datasets containing dense regions, and it was argued that these problems are due to the optimization criterion used in the algorithm for suppression between clones, aimed at eliminating redundancy. They have suggested that it should be feasible to modify such criterion to overcome the problem. However, considering that the criterion was inspired by the Idiotypic Network Theory [16], in that a clone is suppressed

[1] It should noted that this was first pointed out by Dr. T. Stibor during the technical discussions at ICARIS-2007.

if recognized by another clone, such modification may not be straightforward, and may affect the biological inspiration of the algorithm. This motivates us to investigate another modification: the affinity function. As we do not go into details regarding aiNet, the reader is referred to [11] and [15] for details of the algorithm. In addition, we follow the same notation of the parameters used in the algorithm as in [15]. Finally, it should be kept in mind that, because kernel methods can be applied to very general representations (for which a PD kernel is defined), not only vectors, the adapted version of aiNet considered here is not a true kernel method, as it requires a real-valued vector representation.

The aiNet algorithm is based on a set of interconnected antibodies, which represent internal images of antigens to which the network is exposed, modeling the competition for antigenic recognition, while eliminating antibodies that recognize each other. An affinity measure, which, in [11], was originally based on the Euclidean distance, is used to quantify the interaction strength between an antibody and an antigen and, also between two antibodies. The affinity measure results from an adapted distance function, such that the affinity is maximum when the distance is minimum. Therefore, in the following discussion, we consider how such distance function can be modified. Assuming that the input data lie in an input space $\mathcal{X} = \Re^p$, with an antigen $x \in \mathcal{X}$ and denoting a population of n antibodies as $A = \{A_1, A_2, \ldots, A_n\} \subset \mathcal{X}$, the Euclidean distance function is used in three steps in the algorithm:

1. for determining the affinity between an antibody $A_i \in A$ and the antigen x:

$$d_i = \|A_i - x\| \tag{16}$$

2. for mutation of an antibody A_i, generating a mutated clone A_i^*:

$$A_i^* = A_i - \beta \|A_i - x\| W(A_i - x) \tag{17}$$

   where W is a $p \times p$ diagonal matrix, whose diagonal is an independent random vector, with elements independently and uniformly distributed in the interval $(0, 1]$, and β is the hyper-mutation rate. The term $-(A_i - x)$ in equation 17 is the search direction that minimizes the distance between the antibody A_i and the antigen x, and is related to the negative gradient of the square of the distance function 16. The matrix W adds a random component to the direction of search, and, due to its properties, it follows that, for some vector v, Wv lies in the same orthant as v. Therefore, the introduction of diversity (i.e. mutation) in aiNet is achieved through one step of gradient descent with learning rate $\beta\|A_i - x\|$, where the negative gradient is randomly "distorted" by the operator W.
3. for determining the affinity between two antibodies A_i and A_j:

$$d_{i,j} = \|A_i - A_j\| \tag{18}$$

The derivation of the kernel-based version of aiNet follows from simply substituting the Euclidean distance function used in steps 1 and 3, along with the mutation operator employed in step 2, and is based on the EF-KSOM kernel

self-organizing map discussed in [17]. Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \Re$ denote a (PD) kernel function, so that $J_y(x) = \|\phi(x) - \phi(y)\|^2$ is the squared distance in the feature space between the mapped versions of x and y (which can be calculated using equation 5), and let $\nabla J_y(x) = \frac{\partial J_y(x)}{\partial x}$ denote the gradient of $J_y(x)$. For convenience, we consider the use of squared distance measures, so that the three steps previously discussed are given, respectively, by the following equations:

$$d_i = J_x(A_i) \tag{19}$$

$$A_i^* = A_i - \beta\sqrt{J_x\left(A_i\right)}W\nabla J_x\left(A_i\right) \tag{20}$$

$$d_{i,j} = J_{A_j}(A_i) \tag{21}$$

In this formulation, the original implementation presented in [18] can be obtained by using a kernel function $k(x, y) = \langle x, y\rangle$ (so that $J_y(x) = \|x - y\|^2$ and $\nabla J_y(x) = 2(x - y)$), and through an appropriate scaling of the distance-related parameters σ_s and σ_d (suppression and pruning thresholds, respectively) and the hyper-mutation rate β. Then, using equation 5, it can be observed that $\nabla J_y(x)$ can be written as:

$$\nabla J_y(x) = \frac{\partial k(x,x)}{\partial x} - 2\frac{\partial k(x,y)}{\partial x} \tag{22}$$

and mutation is, therefore, performed according to:

$$A_i^* = A_i - \beta\sqrt{J_x\left(A_i\right)}W\left(\frac{\partial k(A_i,A_i)}{\partial A_i} - 2\frac{\partial k(A_i,x)}{\partial A_i}\right) \tag{23}$$

The resultant algorithm then attempts to obtain a population with a non-redundant population of memory antibodies (using the suppression threshold), so that the distance to the input antigens, calculated in the feature space, is minimized. In particular, all distance-related operations (calculation of affinities for the antigen, suppression and pruning) are conducted in the feature space.

Finally, consider the case when a translation invariant kernel function is used, such that the corresponding distance can be written as $J_y(x) = f(\|x - y\|^2)$ (which happens when the Gaussian or Inverse multiquadric kernels are used), where $f : \Re_0^+ \rightarrow \Re_0^+$ is a differentiable monotonic function. Then it is generally possible to obtain values for the distance-related parameters σ_s and σ_p (suppression and pruning thresholds, respectively), which, if used with the Euclidean distance, may lead to the same results as those obtained using such kernel functions. This may happen because the major difference (in addition to a small variation in the number of clones generated during the mutation phase) between using such kernel function or the Euclidean distance is the learning rate, which, in the former case, is equals to $\frac{\sqrt{f(\|A_i-x\|^2)}}{\|A_i-x\|}\left.\frac{df(z)}{dz}\right|_{z=\|A_i-x\|^2}$ times that of the latter, assuming that $A_i \neq x$. Therefore, if this term is approximately equals to one throughout the execution, then it follows that operating in the feature space does not bring anything new to the dynamics of aiNet. Moreover, if the effects of different learning rates are negligible, then the results obtained using either version may be very similar. In the next section, we report on this possible equivalence.

4.2 Experimental Results

To analyze the impact of the use of kernel functions, we consider the experiments reported by Stibor and Timmis [15], used to assess the compression quality of aiNet. Drawing ideas from non-parametric density estimation, they have used a simplification of the Kullback-Leibler divergence (also known as relative entropy) to quantify how close the memory antibodies, returned by aiNet, were to the input dataset.

Let $X = \{x_1, x_2, \ldots, x_{n_x}\} \sim P$ be an input and $R = \{r_1, r_2, \ldots, r_{n_r}\}$ be a reduced dataset, where X is distributed according to some generally unknown probability distribution P, such that $X, R \subset \mathcal{X} = \Re^p$. In the case of aiNet, X and R are the input (antigens) and output (memory antibodies) of the algorithm, respectively. Then, the simplified expression for the calculation of the entropy value is [15]:

$$\tilde{J} = \frac{1}{n_x} \sum_{i=1}^{n_x} \left\{ \ln \left[\frac{1}{n_r} \sum_{j=1}^{n_r} k_h^G(x_i, r_j) \right] - \ln \left[\frac{1}{n_x} \sum_{j=1}^{n_x} k_h^G(x_i, x_j) \right] \right\} \tag{24}$$

where $k_h^G(x, y)$ is the normalized Gaussian kernel with width h:

$$k_h^G(x, y) = \frac{1}{\left(\sqrt{2\pi}h\right)^p} \exp\left(-\frac{1}{2h^2}\|x - y\|^2\right) \tag{25}$$

where a large entropy indicates that R is relatively close to X. Due to the fact that equation 24 is a simplification of the original expression for calculating the Kullback-Leibler divergence, it follows that $\tilde{J}_{X,R}$ can assume any real value, not being, therefore, restricted to non-positive values (as in the case of the Kullback-Leibler divergence). Therefore, by applying aiNet to a given input dataset and calculating 24, it is possible to evaluate the compression quality in a quantitative way. However, due to the fact that the obtained value is a relative quantity, it is necessary to have a reference value, which is used as a baseline for the comparison. In the experiments conducted in [15], as the probability distribution P is known, the reference value was obtained by sampling a reference dataset R_{ref} from P, with cardinality equals to the mean cardinality of the reduced datasets obtained with some parameter set θ. Using this dataset, $\tilde{J}_{ref}$ was used as a reference entropy value, such that when $\tilde{J}$ was close to $\tilde{J}_{ref}$, the compression quality was acceptable.

To conduct the experiments, the same four two-dimensional synthetic datasets used in [15], shown in table 1, were used. The influence of the suppression threshold was evaluated, using the values $\sigma_s \in \{0.5, 0.2, 0.1, 0.05, 0.01, 0.005, 0.001\}$, with the remaining parameters being used as in [15]. In addition to the Euclidean distance, as in the original aiNet algorithm [11], the Gaussian and inverse multiquadric kernels, with $\sigma \in \{0.25, 0.5, 1, 2\}$ (equations 6 and 8), and the polynomial kernel with $c \in \{1, 2\}$ and $d \in \{2, 3, 4\}$ (equation 7) were used in the experiments. Investigation of the equivalence between the results obtained with the Gaussian and inverse multiquadric kernels (see the end of section 4.1) was

Table 1. Synthetic datasets used in the experiments (see [15])

Dataset	Description
1	Gaussian distribution with mean $\begin{bmatrix} 0 & 0 \end{bmatrix}$ and covariance matrix $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
2	Mixture of six Gaussian distributions
3	Mixture of two Gaussian distributions, with both means equals to $\begin{bmatrix} 0 & 0 \end{bmatrix}$ and covariance matrices I_2 and $\frac{1}{4}I_2$
4	Sin/cos-based distribution, composed of two U-shaped distributions

conducted by performing a paired t-test comparing the relative entropies and the memory population sizes. It was found that the only results not different at a 5% significance level were those obtained with Gaussian kernel with $\sigma = 1$ and $\sigma_s \leq 0.1$. The input datasets were taken with 400 points, and each execution of aiNet was repeated 500 times to consider the mean and standard deviation in the results. Finally, to report the results, we present plots of the relative entropy versus the cardinality of the reduced dataset, along with the corresponding reference entropy values. In particular, this graphical representation makes the interpretation of results obtained with multiple distance functions easier, and allows to visualize how the entropy value scales with the cardinality of the reduced datasets.

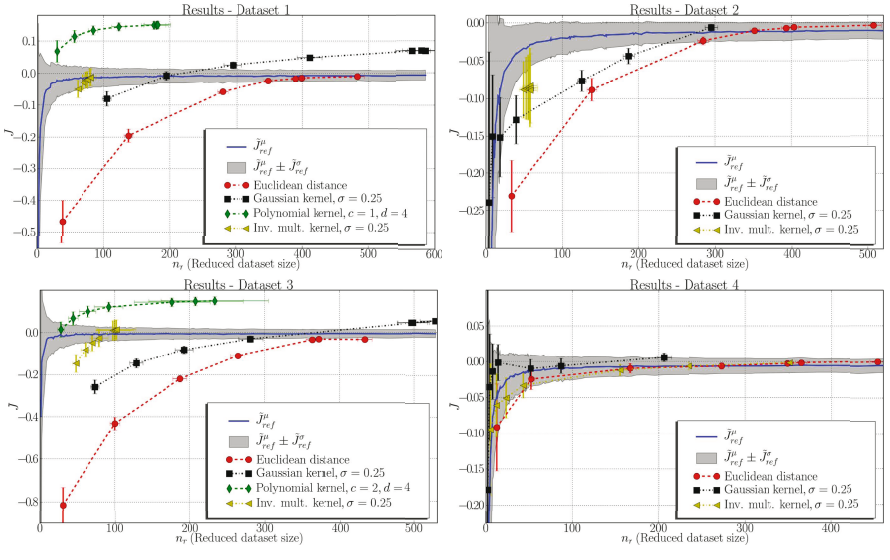


Fig. 1. Obtained results for different distance functions in the four synthetic datasets. $\tilde{J}^{\mu}_{ref}$ and $\tilde{J}^{\sigma}_{ref}$ denote the mean and standard deviation of the reference entropy values, respectively.

The entropy values obtained for the experiments conducted are shown in figure 1, where decreasing the value of σ_s leads to increasing values of n_r, and the shaded regions represent one standard deviation in the reference entropy values. In each graph, only the results obtained with the kernel functions which were superior to those obtained with the original implementation are presented, to make the visualization easier. It can be verified that, in the case of datasets 1 and 3, the results obtained with the Gaussian, inverse multiquadric and polynomial kernels are considerably superior to those obtained with the Euclidean distance, although the Gaussian kernel ($\sigma = 0.25$) can lead to large memory population sizes, especially for small values of σ_s. The inverse multiquadric kernel, with $\sigma = 0.25$, results in relatively small population sizes, varying from 60 to 80 (dataset 1) and 50 to 100 (dataset 3). Finally, in these two datasets, the polynomial kernel with $c = 1$ and $c = 2$, respectively, and $d = 4$ is the one which attains the best results, with positive entropy values superior to the corresponding reference values. However, in the case of datasets 2 and 4, especially the latter, it was difficult to obtain results superior to those obtained with the Euclidean distance. In particular, this was clear for the polynomial kernel, whose entropy values were very low, usually smaller than -1. Upon closer investigation, it was verified that this is related to the numerical properties of the polynomial kernel with un-normalized data, which can be alleviated by the use of small hypermutation rates. For datasets 1 and 3, which are centered around zero, this problem does not arise.

While care must be taken in interpreting some of these results, due to the fact that the approximated entropy values are not non-positive (indicating that the density estimated was closer to the true probability distribution than the density estimated using a dataset obtained from the true distribution), the general trend is that the use of some kernel functions can lead to results better than those obtained with the Euclidean distance. Therefore, we argue that the inadequate results reported in [15] are not only due to the optimization criterion, but appear to be related also to the characteristics of the Euclidean distance in some datasets.

5 Conclusions

In this paper, we have considered the connections between AISs and Kernel Methods. After a brief presentation of some of the theoretical results used in the latter, its applicability in positive/negative selection algorithms, the dendritic cell algorithm and immune networks was discussed. While it is suggested that not much differences should arise in the case of positive/negative selection, kernel functions could have a significant impact in the dendritic cell algorithm and immune network approaches. However, in the case of the dendritic cell algorithm, further studies of its theoretical properties should precede the application of kernel functions. Finally, as a practical application, a kernel-based version of aiNet was presented, and experiments with synthetic datasets were conducted, indicating that, in some cases, properly tuned kernel functions may improve the compression quality of the algorithm.

While the results presented in this paper can be considered as preliminary in the context of the use of kernel functions in immune-inspired algorithms, it may have important consequences, especially in the motivation of future work. As previously discussed, as the kernel function and its parameters define the properties of the feature space induced, these have a major impact on the performance of the algorithms. In this sense, the argument raised by Freitas and Timmis [3], advocating the use of data representations and affinity functions tailored to the target problem suggests an interesting line of research. The theoretical aspects, along with the experimental results reported here inspire the developments of algorithms employing adaptive similarity functions, as suggested in [3]. In the context considered here, such algorithms would be able to adjust the kernel function and/or kernel parameters, allowing for automatically tuned representations, which can be seen as partitioning the input space into regions, and projecting the patterns falling into each region into a corresponding feature space.

Acknowledgments. The authors would like to thank Dr. T. Stibor for providing additional information on the original experiments. This work has been supported by UOL (www.uol.com.br, process number 20060519110414a), FAPEMIG and CNPq.

References

1. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach, 1st edn. Springer, Heidelberg (2002)
2. Perelson, A.S., Oster, G.F.: Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination. J. Theor. Biol. 81(4), 645–670 (1979)
3. Freitas, A.A., Timmis, J.: Revisiting the foundations of artificial immune systems for data mining. IEEE Trans. Evol. Comput. 11(4), 521–540 (2007)
4. Hart, E.: Not all balls are round: An investigation of alternative recognition-region shapes. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 29–42. Springer, Heidelberg (2005)
5. Hart, E., Bersini, H., Santos, F.C.: How affinity influences tolerance in an idiotypic network. J. theor. Biol. 249(3), 422–436 (2007)
6. Stibor, T., Timmis, J., Eckert, C.: On the use of hyperspheres in artificial immune systems as antibody recognition regions. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 215–228. Springer, Heidelberg (2006)
7. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience, Chichester (1998)
8. Schölkopf, B., Smola, A.J.: Learning with Kernels, 1st edn. MIT Press, Cambridge (2002)
9. Timmis, J.: Artificial immune systems - today and tomorrow. Nat. Comput. 6, 1–18 (2007)
10. Timmis, J., Hone, A.N.W., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. Theoretical Computer Science. Theoretical Computer Science (in press) (2008), doi:10.1016/j.tcs.2008.02.011
11. de Castro, L.N., Von Zuben, F.J.: aiNet: An artificial immune network for data analysis. In: Abbass, H.A., Sarker, R.A., Newton, C.S. (eds.) Data Mining: A Heuristic Approach, pp. 231–259. Idea Group Publishing (2001)

12. Esponda, F., Forrest, S., Helman, P.: A formal framework for positive and negative detection schemes. IEEE Trans. Syst. Man, Cybern. B 34(1), 357–373 (2004)
13. Greensmith, J., Aickelin, U., Tedesco, G.: Information fusion for anomaly detection with the dendritic cell algorithm. Information Fusion(in press) (2008)
14. Timmis, J., Neal, M., Hunt, J.: An artificial immune system for data analysis. BioSystems 55, 143–150 (2000)
15. Stibor, T., Timmis, J.: An investigation on the compression quality of aiNet. In: Proc. IEEE FOCI-2007, pp. 495–502 (2007)
16. Jerne, N.K.: Towards a network theory of the immune system. Ann. Inst. Pasteur. Imm. 125C, 373–389 (1974)
17. Lau, K.W., Yin, H., Hubbard, S.: Kernel self-organising maps for classification. Neurocomputing 69, 2033–2040 (2006)
18. de Castro, L.N.: aiNet implementation (2000) (visited in January/2008), ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/lnunes/demo.zip

Boosting the Immune System

Chris McEwan, Emma Hart, and Ben Paechter

Napier University, Edinburgh, Scotland
{c.mcewan,e.hart,b.paechter}@napier.ac.uk

Abstract. Much of contemporary research in Artificial Immune Systems (AIS) has partitioned into either algorithmic machine learning and optimisation, *or* modelling biologically plausible dynamical systems, with little overlap between. Although the balance is latterly beginning to be redressed (e.g. [18]), we propose that this dichotomy is somewhat to blame for the lack of significant advancement of the field in either direction. This paper outlines how an inappropriate interpretation of Perelson's shape-space formalism has largely contributed to this dichotomy, as it neither scales to machine-learning requirements nor makes any operational distinction between signals and context.

We illustrate these issues and attempt to derive both a more biologically plausible and statistically solid foundation for an online, unsupervised artificial immune system. By extending a mathematical model of immunological tolerance, and grounding it in contemporary machine learning, we minimise any recourse to "reasoning by metaphor" and demonstrate one view of how both research agendas might still complement each other.

1 Introduction

Perelson's "shape-space formalism" has become the *de facto* representational abstraction in AIS. Briefly: ligands and receptors are represented as points in an abstract space, with a contiguous region of recognition surrounding each point to account for small differences. Ligands and receptors that have intersecting regions are said to have *affinity*. Although biologically unrealistic, the shape-space has a certain heuristic value in place of the complex bio-chemical process of protein binding.

This abstraction has been adopted wholesale by the AIS community as isomorphic with the vectorial representation of a data-set: each data-point being an artificial *antigen*, perhaps falling under the recognition region of some artificial *lymphocytes*. Whilst pragmatic from a computational perspective, this abstraction both distances algorithms from the underlying biology and essentially reduces to augmented, classical memory-based methods of machine learning; methods that have largely fallen out of favour, because they are unscalable in terms of both the dimensionality and size of the dataset.

AIS algorithms are typically built on very weak immunological analogies, providing no insight into the mechanisms and dynamics they claim to be inspired by. Though there is a tension over what should be abstracted from the biology;

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 316–327, 2008.

often one finds a complete departure from the view of the immune system as an autonomous, self-regulating system. Key ideas are externally and arbitrarily imposed (e.g. self-nonself) and dynamical processes are replaced with trivial simplifications (e.g. threshold based deletion and memory).

We advocate that by trying to better marry key ideas from both contemporary machine learning and theoretical immunology, it should be possible to produce novel contributions relevant to both fields. Therefore, in this paper we attempt a small step in this direction. The paper outlines a biological basis that affords an alternative problem representation, and draws a parallel between the statistical and computational learning concept of boosting [8] and immune learning. Finally, we illustrate how these aspects can be combined in an approach which exploits the uniquely immune notion of *tolerance* to induce an adaptive, nonlinear feature selection and topic-tracking algorithm.

1.1 The *Non*-immune-inspired Foundations of AIS

Memory-based methods (such as nearest-neighbour classifiers and density estimators) are flexible, non-parametric and, as such, have a fairly natural mapping to bio-inspired ideas: the population *is* the model and classification is performed at runtime. By exploiting a sense of *locality* between data-points they tend to be able to fit arbitrary complex decision boundaries or represent dense, nonlinear regions in unlabelled data.

How well memory-based methods perform depends crucially on just how local this sense of locality is. This is where the curse of dimensionality hits hardest: as the dimensionality of the space increases, its volume increases exponentially faster. Any metric defined across this volume becomes increasingly meaningless, as all points tend to become equidistant. The unintuitive effects of the curse have been discussed extensively in the machine learning and statistics literature (e.g. [10,1,2,9]) and an AIS perspective is given by the work of Timmis and Stibor in the context of negative/positive-selection algorithms [17]. However, these undesirable effects have much broader scope, affecting *any* AIS that involves affinity metrics based on recognition-regions in a high-dimensional shape-space. This generally includes idiotypic networks and clonal selection algorithms.

In an n-dimensional shape-space, the immune repertoire must be of the order $O(c^n)$ where c is a constant (e.g. $c = 2$ in a binary space). This exponential scaling is computationally abhorrent for the typically large n involved in machine learning. Even if we assume that the active repertoire is a sparse sampling of the shape-space, an exponential increase in volume means that antigen are also distributed sparsely. Such a sparse sampling is of little practical use, unless we can access exponentially more antigen to fill the void. This further antagonises scalability as memory-based methods must keep all (or a representative majority) of the dataset in memory for runtime performance.

Any AIS that builds on these foundations has some formidable theoretical issues to attend to before justifying any immune inspiration. One might argue that such scaling issues do not effect the real immune system, due to its massive parallelism and high cell turnover. We find this argument unconvincing; the

dimensionality of the biological shape-space is likely orders of magnitude smaller than typical machine-learning datasets[1]. The real problem, is that *ligands-as-data-points* may be an inappropriate abstraction.

2 Biological Inspiration

In [5] Carneiro observed that deriving the dimensionality of a shape-space from empirical protein affinity matrices is an ill-posed problem. Conversely, constructing a theoretical shape-space that is consistent with empirical affinity matrices is equally difficult. In either case, his experiments demonstrate that complementarity is not the only factor in ligand binding: there is a *relational* aspect. Instead, he proposed that (in the meantime) immune models should be robust to the nature of the affinity matrix. In his own work, this took the form of ligands and receptors binding probabilistically, rather than under the ubiquitous, over-simplified geometric lock-and-key metaphor. The result is that affinity is no longer constrained to contiguous, isotropic regions surrounding a point in shape-space – recognition can occur at multiple locations in the space.

Secondly, Carneiro recognised that much of the well-documented paradoxical and limited dynamics of existing immune network models, with respect to inducing tolerance and immunity, stemmed from the symmetry of complementary receptor/ligand interactions [3,4]. Without any method to break that symmetry, he could find no convincing way to induce models that simultaneously captured both the *structure* and *function* of the immune system. By integrating the role of T-Helper cells in activating induced B-Clones (and the secretion of complementary B-Clones in suppressing T-Help), Carneiro could break the symmetry of interactions and realise a model consistent with Varela's *self-affirming network* [19] and Coutinho's *central and peripheral immune system* [16], where tolerance is an active process, dominant over the default immune response.

Carneiro's insights into the relational aspects of an affinity measure and the role of T-Clones in activating the immune response provide the key inspirations for the biological basis of our approach.

2.1 Ligands and Receptors Are *not* Data-Points

Ligand binding is an extremely complex process and active area of research in bio-informatics. It is currently not feasible to produce a biologically realistic model of the protein folding and binding dynamics, and it is not clear that such a model would be of any benefit to machine learning abstractions. However, an often unexplored feature of immune recognition is the different processes in antigen recognition carried out by different immune components. Integrating T-Cells allows us to model these processes and, in turn, redefine our method of problem representation.

Briefly, proteins are long chains of smaller *peptides*; which are themselves, chains of amino acids. Any specific protein has a chain of peptides that make up

[1] E.g. Carneiro estimated ten to twenty dimensions [5].

its *primary structure* – the long chain. The protein then undergoes a complex folding process which results in a three dimensional surface – it's *secondary structure* – where some peptides are buried inside the structure and others brought together on the surface. To quote Janeway [11]:

> Antigen recognition by T-cell receptors clearly differs from recognition by B-cell receptors and antibodies. Antigen recognition by B cells involves direct binding of immunoglobulin to the intact antigen and [...] antibodies typically bind to the surface of protein antigens, contacting amino acids that are discontinuous in the primary structure but are brought together in the folded protein. T cells, on the other hand, were found to respond to short contiguous amino acid sequences in proteins. These sequences were often buried within the native structure of the protein and thus could not be recognised directly by T-cell receptors unless some unfolding of the protein antigen and its 'processing' into peptide fragments had occurred.

To reiterate: T-Cells recognise contiguous aspects of the protein without reference to secondary structure. B-Cells and antibody recognise surface patterns across the secondary structure, without reference to primary contiguity: immunoglobulin binding depends on both abstract proximity in shape-space *and* physical proximity of peptides on the protein surface. This significantly alters the representational and scaling complexity of the immune repertoire and goes some way toward curtailing the curse of dimensionality.

2.2 An Alternative Abstraction

Assuming competitive exclusion, we might expect of the order $O(N)$ viable T-Cells, each able to recognise one of N peptides. If we also assume that a B-Clone can recognise β nearby surface features[2], then we can also expect of the order $O(N(N-1)...(N-(\beta+1)))$ viable B-Cells. This scaling is slower than the polynomial $O(N^\beta)$ in the worst case: a worst case that is only realistic if each peptide is uniformly likely to appear close to another. This is almost certainly false from both the biological and computational perspective as redundancy is typically rife. It is this redundancy that makes learning feasible.

Note that in this form, *recognition is no longer a function of an affinity metric in high dimensional space*. T-Clone populations reflect the density of independently presented antigenic peptides. B-Clone populations reflect the density of peptides that co-occur on the protein surface. This immune system is *feature-driven*. Affinity is down-graded to a low dimensional measure of binding strength and degeneracy between peptide "features", rather than a metric over high-dimensional feature vectors. Recognition now has an implicit temporal and spatial aspect.

Intuitively, the immune system does not classify or cluster protein, but rather learns to tolerate signals that are not indicative of pathology. This recognition

[2] The notion of a fixed β is unnecessary, but simplifies our analysis here.

must be (i) sensitive to differences in morphology, as proteins are structurally flexible and degenerate; and (ii) be sensitive to the context of peptide occurrence, because pathogen are often made of the very same stuff as the self. There is simply no way to integrate these ideas in the traditional kernel-based abstraction of affinity, because it lacks any distinction between signals and context.

3 Statistical Inspiration

We now consider a formalisation of these ideas in terms of online, unsupervised learning; presented in the context of information retrieval and filtering. We somewhat concur with Jerne [12] that language is a reasonable metaphor for the problems the immune system faces, though we intend to traverse that link in the opposite direction.

Given an arbitrary $m \times n$ data matrix A of m features (e.g. words) and n contexts (e.g. documents) there is a limited set of data-derived relations that can be used to induce learning methods. For typical AIS, and memory-based methods in general, these tend to revolve around the $n \times n$ kernel/affinity matrix K that represents measures between contexts (e.g. $K = A'A$). These measures are computed via the pairwise comparison of features (e.g. dot-product, Euclidean distance, co-variance etc) and as such, the scalar measures in K_{ij} are inherently m-dimensional. If m is large, which we will assume it is, any non-trivial metric may be subject to the curse of dimensionality.

By duality, many equivalent results can be derived from the $m \times m$ matrix G that represents measures between features (e.g. $G = AA'$). This is neatly encapsulated in the singular value decomposition, which factorises A into the product of three matrices, derived from the eigen-decomposition of K and G:

$$\begin{aligned} A &= USV' \text{ where} \\ G &= AA' = US^2U' \\ K &= A'A = VS^2V' \end{aligned}$$

and S is a diagonal matrix of the square roots of the shared eigenvalues, U and V are the left and right eigenvectors, respectively. As an example of this duality, a well-known result applied in machine learning is that projection of A onto the basis vectors U (i.e. principal component analysis, see e.g. [10] for details), can be equivalently expressed as a projection of K onto V:

$$\hat{A} = U'A = (S^{-1}V'A')A = (S^{-1}V')(A'A) = \alpha K$$

The significance of this indirection is that certain well-defined modifications of the dot-products in K can be shown to be equivalent to dot-products in an *implied* higher-dimensional space; without the burden of processing the high-dimensional U and A. If the data becomes linearly separable in this implied space, then a standard linear classifier can be induced to perform non-linear classification in the explicit space.

However, this ingenious technique for turning low-dimensional, non-linearly separable data into high(er) dimensional linearly separable data is of questionable utility when the explicit data is already high-dimensional: the implicit dot-product is typically derived from the explicit dot-product or Euclidean distance, *which are already cursed*. Furthermore, K is not conducive to online, continuous learning: it scales in the order $O(n)$ for updating and $O(n^2)$ for storage, where $n \to \infty$.

3.1 Moving Away from High Dimensional Notions of Affinity

For many modern learning and data-analysis tasks, the significant problem is that we have too many dimensions. Furthermore, these dimensions are often not independent, because of e.g. variations in word morphology and usage patterns. Much like the immune-system, *it can be the compound structures that are implicit, not the features they consist of.*

In contrast to the kernel K, G is more suited to online learning. It can be updated in essentially $O(1)$ time[3]. Each insertion or removal of context in A can be efficiently reflected in G by an internal comparison between that context's features. In other words, as a sum of outer-products of the columns of A:

$$G = AA' = \sum_i A_i A_i'$$

which, translated into an online setting, reads:

$$G_{t+1} = G_t + (A_{t+1} A_{t+1}')$$

Note that this online updating is impossible with K as this contradicts the physical nature of incoming context that *contain* features. To perform this type of update with K would require incoming context vectors for each feature, which is logically absurd. Updating K requires an $O(n)$ comparisons between the new and all previous contexts, and any attempt to optimise this by exploiting locality is subject to the curse of dimensionality. This distinction in the duality of matrix analysis is often overlooked when A is considered static and given *a priori*.

The cost of this alternative is that G scales in the order $O(m^2)$ for storage, and m is usually much greater than n. However, the reality of this cost is entirely dependent on the sparsity of G, and the sparsity of G is entirely dependent on what is considered a "context". Because this granularity is implicit in our immune model, we have control over this sparsity. In fact, we might favour a highly sparse representation to satisfy biological plausibility as well as computational tractability: clearly, correlation on the surface of a protein is a much more fine-grained notion of context than correlation on, e.g., the same pathogen. In other words, the immune repertoire may only consider short range correlations, which typically promotes sparsity.

[3] Actually, $O(f^2)$ where f is the number of features present in a given context. For many tasks this is significantly less than the total number of features and is independent of the size of the dataset and its dimensionality.

Essentially, we are stating the immune system is performing some sort of *feature selection* and invite the reader to compare this with the traditional data-vector based interpretation. We now consider what modern machine learning can contribute to this idea of immunological learning.

4 Boosting the Immune System

Boosting [8,10] has emerged as one of the most radical and successful instantiations of the recent trend in machine learning of *ensemble learning*. The general goal of ensemble methods is to improve on the classification performance of a single learning machine, by integrating the results of many diverse learners[4]. The radical aspect of boosting is the formal demonstration of the equivalency of *weak* and *strong* learnability: a weak learner, performing only slightly better than random guessing, can be aggregated into a strong learning algorithm:

$$y_0 = strong(x_0) = \sum_i \alpha_i weak_i(x_0)$$

The relation to basis expansion methods (as discussed above) is also clear:

$$y_0 = \sum_i \alpha_i U_i(x_0)$$

where U_i are the basis "functions". The canonical basis are the eigenvectors derived from G, but this functional form is quite general and powerful: accommodating arbitrary functions of x_0, augmentations of x_0 and any additional per-function parameters. In the case of boosting, each basis is a weak classifier. Intuitively, this can be seen as the integration of base heuristics that will often fail, rather than of fully trained classifiers as employed by ensemble methods in general. We invite the reader to consider an analogy with the immune system, particularly Cohen's *co-respondence*, where coherent system-wide responses emerge from the interactions of unintelligent cells with randomly generated specificity [6]. We also note that there are currently few theoretical models which underpin this concept, despite it often being exemplified in AIS.

The key to boosting's success is that learner diversity is augmented by reweighting the training data and sampling proportionally to these weights. Successfully classified data have their weight decreased, forcing newer learners to compensate their predecessor's bias and concentrate on data that is causing continued classification error. This additional diversity in the data, allows the boosted system to perform *better* than its single best component. Again, we invite the reader to consider an immunological interpretation of these ideas, where antigen population and antibody production are in a constant feedback cycle with antigen presentation and clone activation.

[4] Where diversity can be specified in different ways: different learning algorithms; the same algorithm with different parameters or trained on different subsets of the training data...etc.

4.1 B is for "Basis"

Boosting departs from our immunological ideas insomuch as it is performed in a stage-wise manner. That is, at each iteration, a weak learner is generated and trained on the current data, which is then reweighted based on that learner's performance. This vastly simplifies the fitting of the algorithm's parameters as it does not attempt to fit all parameters in parallel (e.g. as in Neural Networks). Once a learner's parameters are fit, they do not change as new learners are added to the ensemble. The process terminates when a learner can do no better than random guessing.

We propose an immune-inspired augmentation to boosting: replace stage-wise fitting with the meta-dynamics of B-Clone creation and deletion. Intuitively, we can envisage the B-Clone repertoire, with its multi-point recognition, as a massive set of randomly generated basis functions, or weak classifiers. This gives us a rich, and adaptive, method of constructing problem representations from viable sub-components (the basis). Viability is determined by clonal selection of the overlapping repertoire. Representational complexity is then regulated by competitive exclusion over available T-Help.

Note that there is no computationally difficult, parallel parameter fitting in this model. A B-Clone's receptor is simply a trivial, unweighted β-subset of all possible features. In contrast to, say, Neural Networks, these components are simply out-competed and replaced, rather than adjusted to fit incoming data with a desired response.

4.2 T is for "Tolerance"

We alluded earlier that Carneiro's model, based on earlier work by Varela and Coutinho *et al.*, is a model of immunological tolerance. We suggest that in a computational intelligence context, the plasticity and adaptation involved in tolerance may prove to be a more attractive mechanism than isolated notions of immunity based on clonal selection, negative/positive selection and idiotypic memory. Indeed, our model indirectly subsumes each of these ideas.

In our own experiments and extensions of Carneiro's model [13,14] we have highlighted that the model can be induced to perform a sort of *habituation* to antigen that's interaction dynamic with the immune response displays a certain persistence. This is almost certainly true of the ubiquitous and regenerative proteins of the self, and is consistent with the well-known phenomenon of high and low dose tolerance. The essential dynamics of our model can be summarised as this: There is a window of opportunity for an unfettered immune response to antigen, after which the suppressive, tolerance-inducing *anti-response* of antigen-like *complement-of-a-complement* antibody initiates. Once both responses are engaged, the outcome of tolerance and immunity is ultimately decided by the success of T-Help in activating B-Clones that opsonise antigen (i.e. immunity), or anti-B-Clones that suppress T-Clones (i.e. tolerance). Between these extremes, the response dynamics depend largely on the magnitude and dynamics of the antigen (see [14] for further details).

The key mechanism for unsupervised learning is this: as features become tolerised, the topology of G is actively modified, which in turn, partitions the viable

repertoire. This adaptive process is based on very specific criteria of inductive bias. We finish by discussing these criteria and pulling the previous sections into a consistent picture of a learning dynamical system.

5 The Noisy Self

Traditionally, the self-nonself distinction has often been treated as a two-class classification problem, with somewhat arbitrary class labelling based on the application area: e.g. *spam-notspam*. This will not hold under our interpretation. Firstly, our system is based on the correlative and morphological relationships between features, not feature vectors. Secondly, if the self is habituated against, then the self is a very particular type of noise – persistent signals that become perceptually ignored.

This makes some biological and semantic sense. If the immune repertoire is a reflection of the antigenic environment of the host, then one might expect two hosts from the same environment to have similar levels of tolerance toward aspects of self and nonself. These levels of tolerance would roughly subsume both the hosts' position in a *phylogenic* (e.g. antigen unique to human, earlier species and mammals) and *ontogenic* (e.g. antigen unique to western European, British, city dwellers) taxonomy. We find essentially similar requirements in statistical learning, where a feature's discriminatory and aggregatory power is very much problem and dataset specific (e.g. English language documents; about computer science; about algorithms).

Further, it is well known that both protein [15] and term co-occurrence [7] networks follow a roughly power-law degree distribution with high clustering-coefficient. The bell-shaped dose-response curve of the immune system, with its high and low dose tolerance, has an intuitive interpretation in these environments: *weakly connected features are random noise and can be passively tolerated; highly connected features are too ubiquitous to distinguish non-self and should be actively tolerated*. For a power-law distribution, this filters out a massive proportion of the features: the many that make up the distribution's tail, and the few that account for the majority of the peak of the curve.

The bulge in the middle, where an immune response freely develops, presents an opportunity for a non-linear, adaptive method for clustering features, tracking their evolving relationships, and decomposing datasets into base contexts.

5.1 Implicit Learning Via Explicit Tolerance

In a network with a power-law degree distribution and high cluster-coefficient, with passive tolerance to weakly significant features and active tolerance toward weakly discriminatory features, the remaining network quite naturally begins to emphasise, and partition into, cohesive sub-nets of correlated, significantly discriminatory, features (see Figure 1). The exact nature of this process is still under analysis. Though a detailed treatment is outside the scope of this paper, an appeal to intuition is straight forward.

Fig. 1. Snapshot of the evolution of tolerance and immunity on a word-based feature network. **Left**: the magnitude of response is represented by opacity with passive tolerance around the weakly connected periphery; active tolerance (highlighted) in the dense centre; and a gradient of active responses in-between. **Right**: removing tolerated nodes from the network illustrates the decomposition of the network into contextually dependent sub-components.

Each connected component is a loose federation of actively responding B-Cells, bound together by idiotypic interactions via overlapping β-subset receptors. These can be considered as *implicit* higher-order basis functions:

$$y_0 = \sum_k \alpha_k F_k(x_0) = \sum_k \alpha_k \sum_i \beta_i clone_i(x_0) \tag{1}$$

where extracting this implicit basis is a simple process of breadth-first search. This search is of the same computational complexity as Eq. 1, with the exception that it allows us to identify when we transition between different connected components. In other words, the higher-order structure is entirely fluid: resizing, partitioning, and merging as G and the immune repertoire co-evolve.

Note that this strategy is not fitting hyperplanes between clusters, hyper-ellipsoids around clusters, or centroids at the heart of clusters. We are not *a priori* constraining the clusters to a fixed number, a fixed shape, a fixed proportion of the space, or any other arbitrary criteria of *good clusterness* to be optimised. The tolerance mechanism *habituates* to non-discriminatory features in the network, which, in turn, induces partitions of contextually dependent features. These partitions can be used to decompose compound structures (e.g. high-dimensional document vectors) into a sum of low-dimensional base topics that are learned and tracked autonomously.

Clearly, this is not possible when the high-dimensional compound structure is the abstraction of ligand and receptor shape.

6 Conclusion

We have discussed the theoretical and conceptual difficulties surrounding the common AIS abstraction of shape-space. High dimensional kernel-based notions

Table 1. Comparison of conceptual ideas in boosting and immunology

Boosting	**Immune System**
Weak Learner/Basis	B-Clone with β-subset receptor
Strong Learner	Immune repertoire
Weighted majority to increase confidence and reduce variance	System-wide coherent responses via cell co-respondence
Reweighting data to increase accuracy and reduce bias	Feedback between antigen presentation and antibody production
Regularisation to manage representation complexity	Competitive exclusion over available T-Help to focus the repertoire
Stage-wise fitting	Parallel metadynamics
Decision surface e.g. $y \in \{+1, -1\}$	Competing complementary responses

of affinity may be a poor abstraction: they do not scale to large computational intelligence domains; are biologically implausible; and they cannot make an operational distinction between context and signals necessary to realising constructive problem representations in an online setting.

Moving away from high-dimensional kernel-based notions of affinity, we have interpreted the immune system as performing adaptive feature selection via a tolerance mechanism inspired from theoretical immunology. This change of perspective opens up a wealth of previously unavailable theoretical results in computational and statistical learning, most notably, boosting and the strength of weak learnability. We find the correspondence between these theoretical results and common immunological learning ideas quite natural and appropriate (see Table 1). Both perspectives may have something to offer each other, without compromising their individual contributions.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973. Springer, Heidelberg (2000)
2. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1998)
3. Carneiro, J., Coutinho, A., Faro, J., Stewart, J.: A model of the immune network with b-t cell co-operation. i - prototypical structures and dynamics. Journal of Theoretical Biology 182, 513–529 (1996)
4. Carneiro, J., Coutinho, A., Stewart, J.: A model of the immune network with b-t cell co-operation. ii - the simulation of ontogenisis. Journal of Theoretical Biology 182, 531–547 (1996)
5. Carneiro, J., Stewart, J.: Rethinking shape space: Evidence from simulated docking suggests that steric shape complementarity is not limiting for antibody-antigen recognition and idiotypic interactions. J. Theor. Biol. 169, 391–402 (1994)

6. Cohen, I.R., Segel, L.A.: Design Principles for the Immune System and Other Distributed Autonomous Systems. Oxford University Press, Oxford (2001)
7. Ferrer, R., Cancho, I., Sole, R.: The small-world of human language. In: Proceedings of the Royal Society of London (2001)
8. Freund, Y., Schapire, R.E.: A decision theoretic generalisation of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
9. Friedman, J.H.: On bias variance 0-1 loss and the curse-of-dimensionality. Data Min. Knowl. Discov. 1, 55–77 (1997)
10. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, Heidelberg (2001)
11. Janeway, C.A., Travers, P., Walport, M., Schlomchik, M.: Immunobiology, Garland (2001)
12. Jerne, N.K.: The generative grammer of the immune system. Nobel Lecture (1984)
13. Mcewan, C., Hart, E., Paechter, B.: Revisiting the central and peripheral immune system. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628. Springer, Heidelberg (2007)
14. McEwan, C., Hart, E., Paechter, B.: Towards a model of immunological tolerance and autonomous learning. Natural Computing (submitted, 2008)
15. Barabasi, A.-L., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. Nature Reviews Genetics 5, 101–113 (2004)
16. Stewart, J., Carneiro, J.: Artificial Immune Systems and their Applications. In: The central and the peripheral immune system: What is the relationship?, pp. 47–64. Springer, Heidelberg (1998)
17. Stibor, T., Timmis, J., Eckert, C.: On the use of hyperspheres in artificial immune systems as antibody recognition regions. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163. Springer, Heidelberg (2006)
18. Timmis, J., Andrews, P., Owens, N., Clark, E.: An interdisciplinary perspective on artificial immune systems. Evolutionary Intelligence 1(1), 5–26 (2008)
19. Varela, F.J., Coutinho, A.: Second generation immune networks. Immunology Today 12(5), 159–166 (1991)

The Limitations of Frequency Analysis for Dendritic Cell Population Modelling

Robert Oates, Graham Kendall, and Jonathan M. Garibaldi

School of Computer Science, The University of Nottingham
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
rxo@cs.nott.ac.uk
http://www.cs.nott.ac.uk/rxo

Abstract. In previous work we derived a mathematical model which allows the frequency response of a dendritic cell to be predicted. The model has three, key limitations: the model assumes that the intermediate co stimulatory molecule signal is constant; it is only possible to make predictions for a single cell and the model only takes into account the signal processing element of the dendritic cell algorithm, with no attempt to explore the antigen presenting phase. In this paper we explore the original model and attempt to extend it to include the effects of multiple cells. It is found that the complex interactions between the cells creates a one to many relationship between the input frequency and the output frequency. This suggests that traditional frequency-based techniques alone are unlikely to yield an effective automated tuning mechanism.

Keywords: Dendritic Cell Algorithm, Frequency Analysis, Tuning.

1 Introduction

The dendritic cell algorithm (DCA) is a relatively new addition to the field of artificial immune systems (AIS). The DCA can be viewed as a binary decision-making algorithm, for making Boolean choices in uncertain problem environments. Despite being successfully applied to several problems [1,6,7] little work has been carried out to characterise the operation of the algorithm. In [4] the high sensitivity of the algorithm to its input parameters is discussed. However, too little is known to automatically tune these parameters for a given application. Currently trial and error is used to identify the appropriate input parameters for new applications. This can be time-consuming and does not guarantee to find an optimal parameterisation. An automated tuning algorithm would be able to find an good quality set of input parameters for a given application and would provide a good basis for future comparisons between the performance of the DCA and other techniques. To derive such a tuning methodology it is important to mathematically characterise the behaviour of the algorithm. In [8] a mathematical model of a simplified version of the DCA was derived using frequency analysis. This model provides an accurate prediction of what information an individual cell will use to make decisions. The simplified model makes three, key

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 328–339, 2008.

assumptions: the co stimulatory molecule (CSM) signal is assumed to be constant; the model only provides the response of a single cell and the correlation between signals and antigen is assumed to be trivial so is not modelled. A 'trivial' correlation between signal and antigen implies that the delay between an antigen appearing in the system and its affects appearing in the input signals is constant and negligible. It was proposed that such a model would be able to provide a tuning methodology for the DCA's input parameters, based on removing those frequencies that were deemed to contain misleading, noisy data. However, a preliminary tuning methodology based on the original model provided mixed results. The low quality of the tuning results has been attributed to the model's over-simplification of the algorithm.

An extension of the original mathematical model proposed in [8] should yield a more accurate estimate of the algorithm. Such an improvement could be the basis of a superior tuning methodology.

This paper is organized as follows. Section 2 provides an overview of the implementation of the DCA. Section 3 explains how the DCA can be modelled as a filter and the benefits of doing so. Section 4 discusses the limitations of the original model and provides some justification for these assumptions. Section 5 explores an extension of the original model to incorporate multiple cells, the results of which are given and discussed in Section 6.

2 Simplifying the Dendritic Cell Algorithm

The original dendritic cell algorithm is inspired by the biological dendritic cell. As a result many of the original parameters and signals were named after biological signals. For an introduction to the relevant biology, the interested reader should refer to [1,4]. In [8] a simplified version of the algorithm was presented which reduced the amount of processing carried out per cell. This optimisation also provides the basis for later extensions that make the frequency analysis of the algorithm possible.

2.1 The Original Dendritic Cell Algorithm

A full description of the original DCA is outside the scope of this paper. The interested reader is referred to [2] for pseudocode and a detailed description of the algorithm's implementation. In this section we provide a brief overview of the operation of the algorithm. The DCA attempts to assign a value to each input symbol between 0 and 1 that describes the likelihood that the antigen is a member of a target set. The algorithm has two parts: a decision making process and a state correlation process. A block diagram of the original dendritic cell algorithm is given in Fig. 1. The algorithm receives four inputs from the problem environment, a stream of enumerated symbols (termed 'antigen') and three normalised signals generated by application-specific heuristics. The output of the algorithm is a stream of enumerated symbols, each associated with a score between 0 and 1. The score is the algorithm's 'decision' about that symbol. The meaning of the

decision depends on the application that the algorithm is being applied to and the input heuristics being processed. The input enumeration stream provides an asynchronous list of symbols which represent the state of the problem environment. The three input heuristics convey the pertinent information for decision making from the problem environment to the dendritic cell population. These signals are expressed as real-numbers. Min-max normalisation is used to keep each heurisitic output within the range 0 and 1. The *PAMP* heuristic identifies situations that *only* occur when a positive output is required. The *Safe* heuristic identifies situations that *only* occur when a negative output is required. The nature of uncertain decision making environments means that these are rarely the inverse of one another. The *Danger* heuristic identifies situations that *always* occur when a positive decision is required, but *can* occur when a negative decision is required. All information that is provided to the algorithm from the problem environment is stored in a collection of asynchronous buffers termed 'tissue'. In the decision making element of the algorithm, the cells accumulate three internal signals based on weighted sums of the input signals. These internal signals are all real-numbers. The IL-10 signal increases proportionally to the *Safe* signal. The IL-12 signal increases proportionally to the *PAMP* and *Danger* signals, but can be decreased by the *Safe* signal. The *CSM* signal increases proportionally to the sum of all signals. When the accumulated *CSM* signal in a given cell reaches a cell-specific migration threshold, the cell makes a decision. If the accumulated IL-10 signal is greater than the accumulated IL-12 signal, the decision is negative. Otherwise the decision is positive. During the sampling life of the cell, it also collects samples of the symbols presented by the enumeration stream. The algorithm can be run continuously in real-time as when a cell finishes its sampling phase, it is removed from the population and a new cell is put in its place, maintaining a constant population of sampling cells. The state correlation element of the algorithm performs statistical analysis on the symbols collected by each cell and each cell's output decision. This correlation is designed to spot patterns between periods of signal activity and the presence of certain antigen.

A crucial factor in the performance of the algorithm is the probability distribution used to allocate migration thresholds to the cell population. If set too high the cells in the population will spend a large amount of time collecting antigen samples before making a decision. This means that correlation becomes an intractable problem as all cells will contain samples of almost all antigen.

If set too low, cells will be vulnerable to noise and in applications where there is a lag between antigen presentation and signal generation, the correlation process will fail.

2.2 Similarities to Neurons

The computational implementation of dendritic cells and neurons both involve performing weighted sums of input signals which are ultimately thresholded to produce a binary output. However, there are key differences between this algorithm and perceptrons. Firstly, perceptrons require supervised training periods to calculate the weightings for a given application, while the DCA uses expert

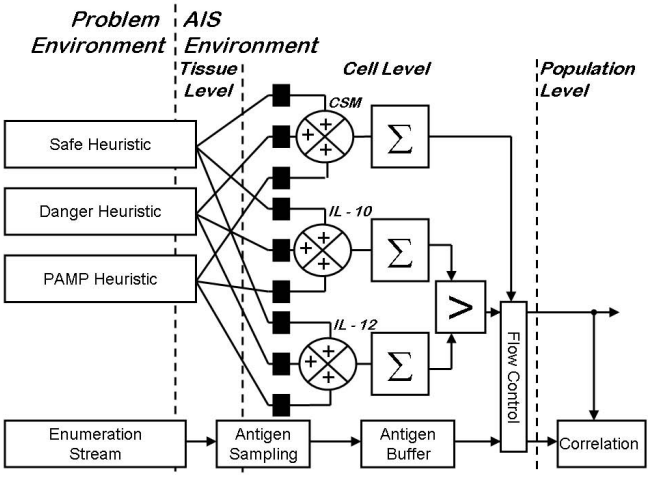


Fig. 1. A representation of the original DCA. The decision is made when the cumulated *CSM* is greater than the cell-specific migration threshold. The decision is positive if the cumulated IL-12 signal is greater than the cumulated IL-10 signal, otherwise the decision is negative.

knowledge to form the input heuristics. The pros and cons of supervised learning vs. encapsulation of expert knowledge are beyond the scope of this paper. Secondly, perceptrons form $N-1$ dimensional hyperplanes, (where N is the number of inputs) and perform linear thresholding using that hyperplane to make decisions. The output from each dendritic cell can also be viewed as the result of applying a hyperplane threshold to the signals that it has been exposed to. However, the dimensionality of the hyperplane is a function of signal strength, as the number of samples taken before migration is determined by the accumulation of the *CSM*. Finally, the DCA processes both signal and antigen over varying size time windows which is not the case with a perceptron.

2.3 The Optimised Dendritic Cell Algorithm

By rearranging the block diagram in Fig. 1 it is possible to make improvements to the performance of the DCA. The final comparison between IL-10 and IL-12 can be replaced by comparing the difference between the two signals with zero. As the two signals are both weighted sums of the same three input signals, the instantaneous difference between the two can be expressed as a single weighted sum. The new abstract signal is termed K. Finally, all calculations that require no persistent state can be calculated per population rather than per cell, significantly reducing the number of calculations required to implement the algorithm. The optimised algorithm is illustrated in Fig. 2. It is estimated that for a population of 100 cells this reduces the number of operations per iteration from 180 (3 multiplications and 3 additions for 3 signals per cell) to 12

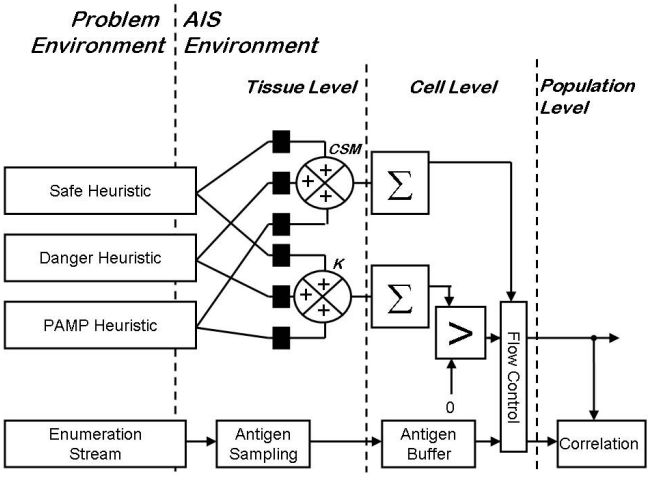


Fig. 2. A representation of the optimised DCA. More processing has been moved into the tissue and fewer calculations are required for the intermediate signal generation.

(3 multiplications and 3 additions for 2 signals per cell). This estimate is only based on arithmetic operations, not assignment operations.

3 Modelling the Dendritic Cell Algorithm as a Filter

It is hoped that by modelling the DCA as a filter it is possible to gain an insight into the workings of the algorithm that will make automated population tuning possible. In the optimised version of the DCA, the tissue now provides 2 input signals to the DC population, K and CSM. These signals are both weighted sums of the input heuristics. K represents the information used to make the decision and CSM is a control signal which affects how long the cell will remain sampling K. The tolerance of the algorithm to noise, as discussed in [7], suggests that not all frequencies of the K signal are processed by the DC population. In order to gain some insight into which frequencies are used and which frequencies are ignored, it is possible to reconstruct the K signal from samples taken by the DC population. Comparing the magnitudes and frequencies of the reconstructed signal $\hat{K}$ and the input signal, K allows a model to be produced of what information is passed through the cell. Note that this estimation of $\hat{K}$ is not a suggested extension or improvement to the algorithm, merely a tool to analyse the standard algorithm. In order to estimate $\hat{K}$ it is necessary to keep track of how long each cell samples for. By dividing the accumulated K signal by the length of time each cell samples for it is possible to estimate $\hat{K}$ for a given cell. The full derivation of the model can be found in [8]. The final result, relating migration threshold M_i, the constant CSM signal, C and the frequency response of the cell is given by:

$$H(\omega) = \frac{\sum_{g=0}^{W_L-1} \sum_{b=0}^{W_L-1} e^{-jb((\omega+(2g\pi)))}}{W_L^2} \quad (1)$$

where ω is the frequency of the input signal, j is the square root of -1 and W_L is defined as:

$$W_L = \left\lceil \frac{M_i}{C} \right\rceil \quad (2)$$

3.1 Verification of the Model

To verify the model, the response generated by the model was compared to the output of a DC. To measure the frequency response of a DC, sine waves with an amplitude of 1 were presented as inputs, at varying frequencies. The maximum magnitude of the output was used as an approximation of the gain of the cell for each frequency. Figs. 3 and 4 show the results of these experiments. In every case the model is evaluated in the range from 0Hz to half of the sampling frequency of the system. The Nyquist frequency of any system is $\frac{1}{2}f_s$ where f_s is the sampling rate. It is only necessary to examine the system within this range, as accuracy up to the Nyquist frequency guarantees the same level of accuracy for all frequencies. For details please see pages 41-43 of [5].

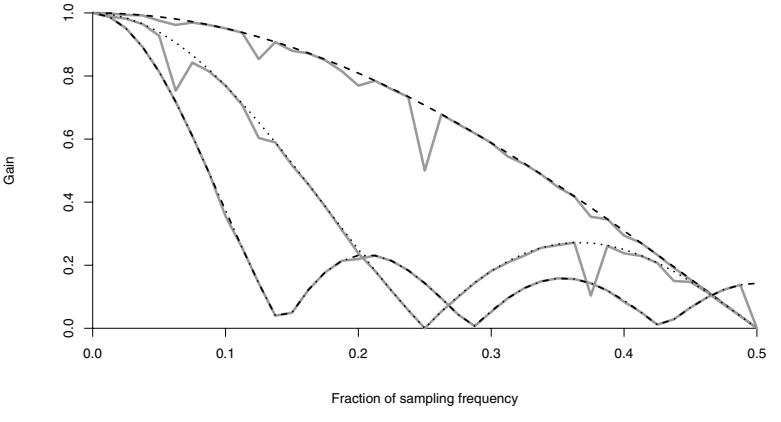


Fig. 3. The Effects of Varying the Migration Threshold. For these experiments the value of the *CSM* is held at 20 and the migration threshold is 30 (dashed line), 60 (dotted line) and 120 (dot-dashed line0). In each case the corresponding actual response is shown as a solid grey line. Data taken from [8].

The model predictions are reasonably accurate across the range of input parameters. Some transient drops in the algorithm's response are not predicted but the general shape of the response is well matched.

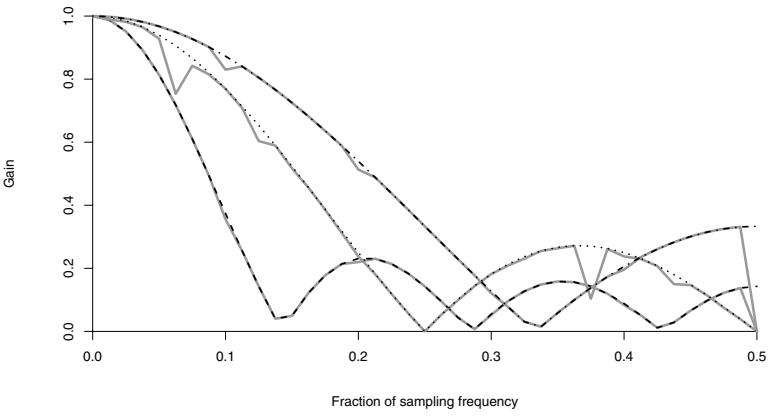


Fig. 4. The Effects of Varying the *CSM* Value. For these experiments the value of the migration threshold is set to 60 and the *CSM* value is 10, (dashed line) 20, (dotted line) and 30, (dot-dashed line). In each case the corresponding actual response is shown as a solid grey line. Data taken from [8].

4 Limitations of the Frequency Model

The assumptions made to derive the model limit how useful the results are for predicting the response of the DCA. Here we discuss the key assumptions and the effects that these assumptions have on the model.

4.1 Constant Co Stimulatory Molecule (*CSM*)

The model assumes that the *CSM* signal is kept constant over the lifetime of the cell. This is unlikely, as the K signal and the *CSM* signal are both weighted sums of the same three input signals, so whilst it is possible for one to move independently of the other, it is highly unlikely. However, it is doubtful that this is a factor in the model's accuracy. The *CSM* signal is accumulated by a cell over its lifetime. This means that any constant model of *CSM* is equivalent to any selection of the *CSM* signal with the same accumulated total over the lifetime of the cell. The implication of this is that the model allows the user to inspect the cell's behaviour for a small range of *CSM* values. Thus any tuning methodology based on this model would be valid if the selected *CSM* was a good representation of important regions of activity for the application.

4.2 Antigen Correlation

The model makes no attempt to take into account the antigen correlation of the algorithm, so it can make no predictions about how this element of the algorithm is effected by the input parameters. For applications where the correlation between antigen presentation and signal presentation is trivial this is unimportant.

For example, if there is no delay between the antigen being presented and its effects being felt, the results of the model would be adequate for representing the application's needs. However, when there is a delay between antigen presentation and the resulting signal presentation, or where the relationship between antigen presentation and signal presentation is combinatorial, (i.e. no one antigen is responsible for a positive decision, but certain combinations of antigen can cause this to happen) the model will not provide enough information to select the migration threshold range. It is of note that there are no applications of the DCA in the literature where combinatorial effects have been investigated. The model could be used in the future to investigate the effects of M_i on cases where there is a time delay between antigen presentation and signal presentation, as the phase of the $\hat{K}$ signal will provide information about the lag introduced by the algorithm and thus, the largest possible time between sampling an antigen and ceasing to process signal.

4.3 Single-Cell Modelling

The model only considers a single cell operating in isolation from the rest of the population. This is considered to be the most significant drawback to the practical use of this model for migration threshold tuning. The DCA relies on the use of a population of cells to ensure that samples are processed frequently and to gather a wide range of data from multiple frequencies. By ignoring the interaction between a population of cells it is likely that the model is an oversimplification. For this reason it was decided to extend the model to incorporate multiple cells.

5 Extending the Frequency Model

In order to model multiple cells in the frequency domain, it is necessary to specify how they will interact during the normal operation of the algorithm. To produce a population-wide $\hat{K}$ we must find a reliable way of combining the data from a population of cells. For the purposes of this investigation it was decided to simply periodically sample the cell population and check for migrated cells. The $\hat{K}$ output from each migrated cell would be averaged together to produce a population-wide estimate of K for that window. By averaging together the output from multiple cells, the process of generating a multi-cell model is made much easier. In the frequency domain, the averaged output from multiple filters can be modelled as simply the sum of the gains. The averaging process has no effect on the shape of the response, but scales it to be in the range 0-1. To explore the effects of this multi-cell model a 2 cell system was created using one cell with a migration threshold of 90 and one cell with a migration threshold of 110. The *CSM* signal was held at 20 and the sampling rate was held at 1Hz. The output of the cells was checked every algorithm cycle. All of the experiments were carried out using the Octave environment. Fig. 5 shows the frequency responses for the single cell models using an M_i of 90 and 110 respectively.

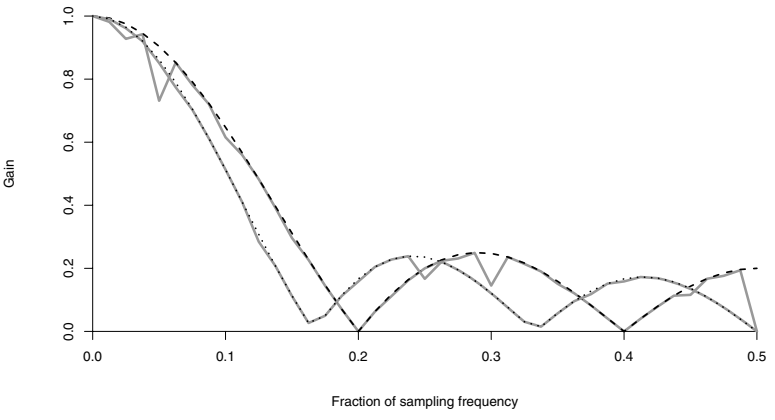


Fig. 5. The frequency responses of the single cell models for a migration threshold of 90 and 110. The dotted-line illustrates $M_i = 90$ and the dashed-line illustrates $M_i = 110$. In both cases the corresponding solid, grey line is the actual response.

6 Results and Discussion

Fig. 6 shows the frequency response of the actual system and the predicted output. The two lines clearly diverge more than the other models. The source of the difference is a combination of the asynchronous nature of the dendritic cell algorithm and the way in which the actual system gain is calculated. To calculate the gain of the actual system, the peak value of the output is recorded by the simulator. As the cells have different migration thresholds there will be occasions when one cell reports and the other does not. On other, rarer occasions,

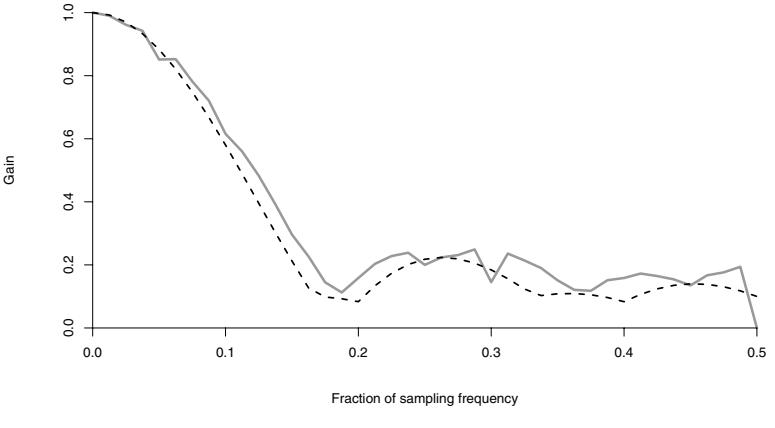


Fig. 6. The frequency response of the two cell system. The dashed line is the predicted response and the solid, grey line is the actual response.

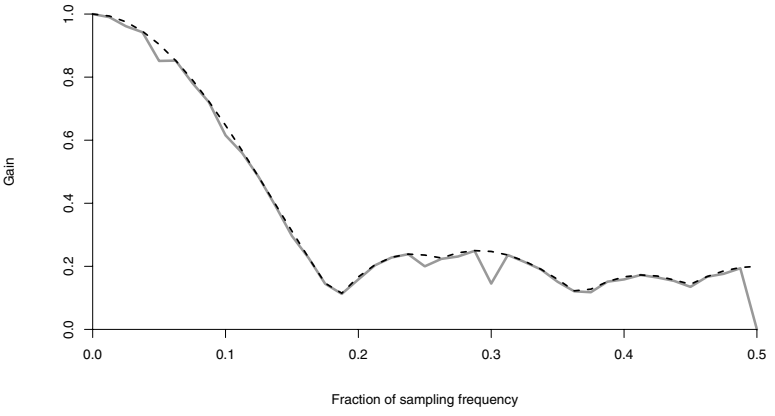


Fig. 7. The frequency response of the two cell system. The dashed line was generated by using the largest gain out of the two, single cell predictions for each frequency. The solid, grey line is the actual response.

both cells will synchronise and report at the same time. As the maximum peak is recorded as a measure of gain, the cell with the larger gain for that frequency will dominate the results from the simulator. This can be verified by comparing the measured response from the algorithm with the maximum of the two single-cell model predictions. In Fig. 7 the output from the actual system clearly follows

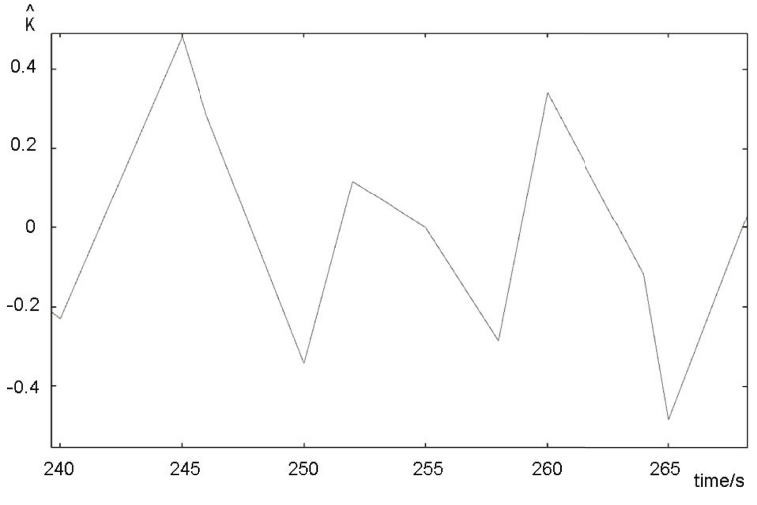


Fig. 8. An example of the output for a two-cell DCA. The sample frequency is $1Hz$ and the input frequency is a sine wave at $0.125Hz$ with a magnitude of 1. The first peak is the gain of the cell with a migration threshold of 90 (approximately 0.48), the second peak is the gain of the cell with a migration threshold of 110 (approximately 0.31) and the third peak is the average gain of each cell (approximately 0.40).

the maximum path of the two model predictions. Fig. 8 shows an example of the asynchronous system outputting three different sized gains for a single input frequency.

The construction of a model capable of predicting the response of a population of DCs is a non-trivial task. The asynchronous nature of the population means that the differing phases of the cells will have a significant effect on the output of the system. Effectively the relationship between gain and input frequency has ceased to be expressible using conventional means, as the gain for a given frequency is a range of values, depending on the relative phases of the cell population. For a two cell system there are four possible gains for each frequency, the gain of cell 1, the gain of cell 2, the average gain of cell 1 and cell 2 and a gain of zero, when neither cell migrates. It is possible to derive that the number of possible gains for a single input frequency, for a population of cells is given by:

$$N_g = 2^P \tag{3}$$

where N_g is the number of possible gains and P is the number of cells in the population. This is a worst-case that assumes that it is possible for all cells to simultaneously drift in and out of phase with one another. For a standard 100 cell implementation of the DCA this evaluates to approximately 1.27×10^{30}. Whilst it is possible to calculate the average response, it is questionable if this will be sufficient to provide enough information to effectively tune the system. It is possible that the cells drifting in and out of phase with one another adds another level of filtering to the system. A transient spike will be picked up by some, but not all of the cells migrating at a given interval, thus the average output over the population will potentially remove some of the noise from the inputs.

7 Conclusions and Future Work

These results cast doubt on the usefulness of traditional frequency-based techniques for modelling the DCA. An effective, multi-cell model, potentially needs to be able to take into account the differing phases of the cells, but even for standard implementations the space of possible gains is huge. The average response could be calculated with knowledge of how often combinations of cells drift in and out of phase with one another. This is calculable for a constant *CSM* system by using the different values of W_L. Such a model would only be a guideline for the general case of the algorithm and the computational complexity of evaluating such a model could potentially outweigh the benefits of automated parameter tuning vs. the trial and error approach.

Acknowledgements

The authors would like to thank Phil Birkin for his advice and input. This work is financially supported by MobileRobots Inc.

References

1. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing dendritic cells as a novel immune inspired algorithm for anomaly detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627. Springer, Heidelberg (2005)
2. Greensmith, J., Aickelin, U., Twycross, J.: Articulation and clarification of the dendritic cell algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163. Springer, Heidelberg (2006)
3. Greensmith, J., Twycross, J., Aickelin, U.: Dendritic cells for anomaly detection. Congress on Evolutionary Computation (CEC) (2006)
4. Greensmith, J.: The Dendritic Cell Algorithm. PhD Thesis. The University of Nottingham (2007)
5. Ifeachor, E.C., Jervis, B.W.: Digital Signal Processing: A Practical Approach. Prentice-Hall, Englewood Cliffs (2001)
6. Kim, J., Bentley, P.J., Wallenta, C., Ahmed, M., Hailes, S.: Danger is ubiquitous: Detecting mis- behaving nodes in sensor networks using the dendritic cell algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, Springer, Heidelberg (2006)
7. Oates, R., Greensmith, J., Aickelin, U., Garibaldi, J., Kendall, G.: The Application of the Dendritic Cell Algorithm to a Robotic Classifier. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 204–215. Springer, Heidelberg (2007)
8. Oates, R., Kendall, G., Garibaldi, J.M.: Frequency Analysis for Dendritic Cell Population Tuning. Evolutionary Intelligence 1(2), 145–157 (2008)

Empirical Investigation of an Artificial Cytokine Network

Mark Read[1], Jon Timmis[1,2], and Paul S. Andrews[1]

[1] Department of Computer Science, University of York, UK, YO10 5DD
{markread,jtimmis,psa}@cs.york.ac.uk
[2] Department of Electronics, University of York, UK, YO10 5DD

Abstract. We present an empirical investigation of a mathematical model of cytokine and cell interactions, as derived by Hone and van den Berg, with the intention of ascertaining its potential for deployment in an engineering context. The model's behavioural robustness regarding various patterns of input is assessed, as is it's scope for manipulation through parameter adjustments. Based on these observations we make an addition to the system and attempt to deliberately engineer particular behavioural properties. Finally, the system's suitability for use in engineering is assessed.

1 Introduction

Hone and van den Berg [1,2] presented and Artificial Cytokine Network (ACN), a mathematical framework for the modelling of cytokines. Their framework utilises ordinary differential equations, and proposes mechanisms through which cells, cytokines, and external stimuli can be modelled and interact. We are interested in the potential application of the ACN in an engineering context. Stepney et al [3] propose a structured framework for the exploitation of biology in the derivation of biologically inspired algorithms. We adhere to this framework's methodology, and present here an initial empirical investigation of the ACN.

Our paper is structured as follows. In Section 2 a numerical instantiation of the ACN is introduced (hereafter referred to as 'the system'), this is used as the basis for our investigations. Section 3 details those properties of any system that we deem critical for deployment in an engineering domain. Section 4 reports the ACN's behaviour when perturbed with a variety of input patterns. In Section 5 we investigate the behavioural impacts of varying the system's parameter values. Section 6 details our proposed additions to the system, and we attempt to deliberately engineer particular behaviours. In Section 7 we assess the system's suitability for deployment in an engineering context. Finally, we conclude our work in Section 8.

2 Simple Cytokine Network

Hone and van den Berg's work [2] presented a general framework for the instantiation of particular cytokine networks. The number of cytokines, cell, sources

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 340–351, 2008.

of external stimuli, and their interactions may all be varied between instantiations, and the framework comprises a large number of parameters. The scope for complexity is huge and an exhaustive study is not feasible, hence we adopt the relatively simple 'numerical instantiation' presented in Section 4 and Figure 4 of [2] as our baseline for these investigations.

The system comprises variable concentrations of a single cell type v, and two antagonistic cytokine types u_1 and u_2. Their dynamics are defined by the following equations (we retain the symbols of the original paper).

u_1 concentration change	$\dot{u_1} = \psi_1(u_1, u_2, s)v - \nu_1 u_1,$
u_2 concentration change	$\dot{u_2} = \psi_2(u_1, u_2, s)v - \nu_2 u_2,$
cell concentration change	$\dot{v} = (\phi(u_1, u_2, s) - \mu)v,$
cytokine secretion	$\psi_j(u_1, u_2, s) = \bar{\psi_j} S(\sum_{k=1,2} W_{jk} u_k - \tilde{\theta_j}),$
cell proliferation	$\phi(u_1, u_2, s) = s\, u_2\, exp(-\gamma\, u_1),\ \gamma > 0$
sigmoid function	$S(x) = 1/(1 + exp\{-x\})$

ψ_j denotes the secretion rate per cell of cytokine j, and is influenced by the concentrations of cytokines and input stimulus in the system. Cytokine concentration increase through secretion is offset by decay, defined as a fixed proportion of the population by ν_j. Similar mechanisms control cell proliferation, ϕ representing the number of offspring a single cell produces, which is offset against a decay rate μ. The maximum level of cytokine j secretion by any single cell is denoted by $\bar{\psi_j}$. $S(x)$ is a sigmoid function that determines what proportion of $\bar{\psi_j}$ is secreted. W_{jk} represents an interaction matrix that denotes the effect that the presence of cytokine k has on the secretion of cytokine j. Threshold values $\tilde{\theta_j}$ must be overcome before significant secretion of cytokine j can commence. Input stimulus s and presence of cytokine u_2 promote cell proliferation, while u_1 hinders it. γ scales the effect of u_2.

All the system's parameters are listed in Table 1. During the course of the experimentation carried out below, parameters take the default values in this table unless otherwise stated. $u_1(0)$ designates a value at the start of the simulation. The interaction matrix W is set as follows:

$$W = \begin{pmatrix} W_{1\,1} & W_{1\,2} \\ W_{2\,1} & W_{2\,2} \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}$$

Hence, the presence of u_1 strongly discourages its own secretion, presence of both u_1 and u_2 promote each others secretion, and the presence of u_2 has no effect on its own secretion.

The system can be perturbed by manipulating the amplitude and temporal properties of the input stimulus s. The change in cell concentration is interpreted as the system's output. It is the relations between inputs and outputs that form the behaviours we investigate.

3 Criteria of Investigation

The following system properties and behaviours are deemed desirable for deployment in an engineering context.

Table 1. Parameters, and their default values. Taken from [2], Section 4 and Figure 4. The 'standard system' is a system with all parameters configured according to this table.

Parameter	Value	Parameter	Value
$u_1(0)$	6.5	$\tilde{\psi_1}$	1
$u_2(0)$	12.5	$\tilde{\psi_2}$	1
$v(0)$	12.5	$W_{1\,1}$	-1
ν_1	2	$W_{1\,2}$	1
ν_2	1	$W_{2\,1}$	1
μ	10	$W_{2\,2}$	0
$\tilde{\theta_1}$	6	γ	0.1
$\tilde{\theta_2}$	11		

Responsivity towards diverse inputs. A system that is reactive towards a wide variety of inputs will be suitable for a diverse set of engineering domains.

Behavioural stability given input instability. The system should exhibit predictable, stable, and recoverable behavioural changes to anomalies in its input stream, rather than becoming unstable and erratic.

Scope for behavioural change. It is important that the dynamics of the system can be altered. A system with one fixed behaviour will not be applicable to a diverse set of problem domains.

Scope for predictable behavioural adjustment. It is important that the system can be tailored towards a particular problem domain, thus we desire that varying parameters have a reliable and predictable effect on the system's behaviour.

4 System Response to Input Patterns

The experiments carried out by Hone and van den Berg [2] utilised either a continuous, or only two distinct inputs. It is fair to assume that in any online scenario the system will be perturbed with multiple, asynchronous input stimuli which vary in amplitude. We wish to ascertain the standard system's stability and responsiveness to these more complex types of input pattern.

Though not as complex as a potential online application, the experiments that follow should collectively provide an indication of the system's response to more complicated patterns. It should be noted that in these simulations the system is always started with an input present. For convenience the period, duration, and amplitude of input stimulus s are denoted P, D, and A respectively. Unless otherwise stated, $D = 0.2$ time units in all cases.

Square Wave Inputs

The system remains responsive to inputs of $P \leq 2.3$ and $A = 8.0$. For periods greater than, or amplitudes less than, these values, system wide decay of all

variables ensues. Conversely, larger A will extend the maximum period for which the system remains responsive, and smaller P will remain responsive despite smaller A.

As P approaches $D = 0.2$, the height of cellular proliferation peaks decreases, and the height of proliferation troughs increases. For $P = D$, i.e. constant input, all variable concentrations experience initial fluctuations but converge on constant values. Maintaining $P = D$, $A \leq 3.2$ induces system decay, whilst larger values increase the post-fluctuation concentration levels.

Very small periods of $P > 0.2$ produce cell proliferation peaks for every input event. For $0.5 \leq P \leq 1.2$ proliferation occurs for every other input, the remaining inputs inducing no response.

Square Waves with Linear Amplitude Increase
The input pattern utilised here has $P = 2.0$, $D = 0.2$, and an initial $A = 8.0$ which is increased by some constant c with each successive input. For all constants c, the system reacts to the increasing stimulus with approximately linearly increasing concentrations. However, when the inputs reach an amplitude of around 50 the proliferation peaks become erratic.

This phenomenon can be attributed to the nonlinear influence that u_1 exerts on the system. There will come a point (when $A \approx 50$) when sufficient u_1 still resides in the system so as to prevent any cell proliferation from taking place when an input occurs. This in turn prevents cytokine concentration increase, and when the next input arrives the cytokine levels will be abnormally low resulting in an extremely large spike.

Square Waves with Exponential Amplification
The system's reaction to exponential increase in input amplitudes are similar to those of linear increase. Increases in peak size correlating to the increase in input amplitudes are observed, and erratic behaviour ensues once input magnitudes of over 50 are reached.

Square Waves with Amplitude Anomaly
This input pattern is identical to the standard square waves above ($P = 2.0$, $D = 0.2$, $A = 8.0$), save for a single anomalous input whose amplitude can can be varied. The experiments here are to examine how the system reacts to an anomaly in an otherwise regular and stable series of inputs. The anomaly is set to occur when the system has settled into a stable pattern.

An anomaly amplitude of 4.0 causes the system to decay immediately. An amplitude of 5.0 causes some instability following the anomaly, and system decay follows shortly. An Amplitude of 5.3 allows system resuscitation, though it requires time to return to the original stable pattern. Large anomaly amplitudes do not cause decay, the system's reaction is a large spike, which if sufficiently large causes temporary instability as result of the considerable quantities of u_1 generated.

5 Parameter Investigations

This Section investigates the effect of varying parameters. Unless otherwise stated, parameter settings are fixed to those values detailed in Table 1, and the input pattern used is the standard square wave detailed above, with $P = 2.0$, $D = 0.2$, and $A = 10.0$. This input pattern is chosen because it induces 'typical behaviour' for the standard system, which is depicted in Figure 1.

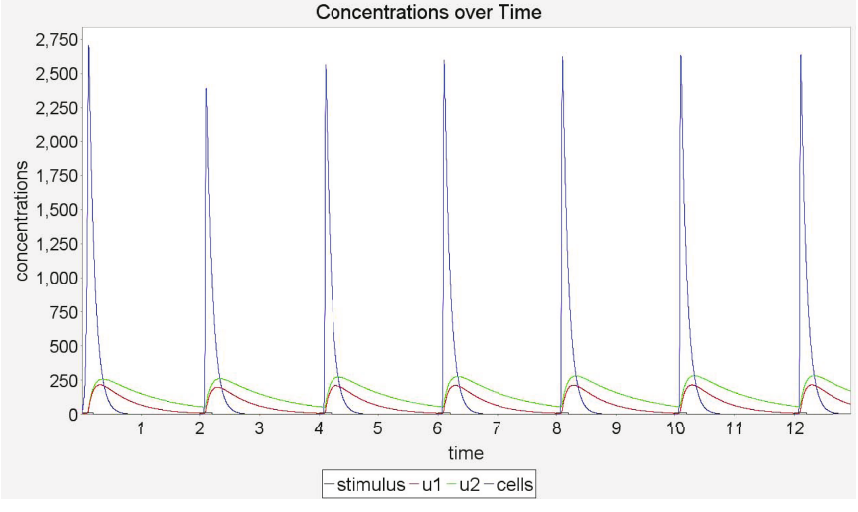


Fig. 1. Systems response to square wave of period 2.0, duration 0.2, amplitude 10.0

Parameters: $u_1(0)$, $u_2(0)$, and $v(0)$.
Effects: If chosen incorrectly, i.e. u_1 too large, or either u_2 or v too small, system wide decay of variables ensues. Otherwise, these variables adjust the instability of the system at time 0. Sufficient disequilibrium will cause large fluctuations in all variable concentrations before the system settles. These variables have no bearing on system behaviour beyond initial instability.

Parameters: ν_1, ν_2, and μ.
Effects: The system is quite sensitive to alterations of either ν_i decay rates. To instil a stable robust behaviour ν_1 must hold a value roughly double that of ν_2. The exact relative difference dictates the quantities of each cytokine that linger following an input, which affects the minimum period to which a response is induced by every input, and the heights of cell proliferation peaks. Given a fixed P , too small a difference induces system wide decay through higher persisting concentrations of u_1. Too large a difference and the system behaves erratically, with small u_1 concentrations allowing huge cellular proliferations, which in turn generate sufficient u_1 to suppress the inputs that follow.

μ adjusts the gradient of proliferation peaks, with small μ creating flatter 'sawtooth' like patterns and very large μ inducing system wide decay, since cells decay at such a rate that non remain to proliferate. Thus, μ adjusts the maximum period to which the system remains reactive, and the minimum period for which each individual input induces proliferation. Note that the effects of μ, ν_1, and ν_2 are linked; variable persisting cell concentrations induced by adjusting μ will affect cytokine concentrations through secretion, which in turn affect the number of cells that proliferate as a result of inputs. Hence, adjustments to μ change the height of proliferation peaks; smaller values will generally reduce heights. However, if alternate input reactions are suppressed by high levels of lingering u_1, the height of the remaining peaks will be significantly greater.

Universally scaling values of ν_1, ν_2, and μ allows cytokines and cells to linger in the system for adjusted periods of time. This can alter the periods for which responses can be mounted for every input, and the maximum period for which the system does not decay. Note that uniformly scaling these parameters does not maintain peak heights.

Examples: If $\nu_1 < 1.8$ system wide decay ensues, with smaller values inducing more rapid decay. $\nu_1 = 3$ provides peaks that are an order of magnitude larger than those produced by $\nu_1 = 2$. Values of $\nu_1 > 4$ causes notable initial instability, and alternate proliferation peaks have an order of magnitude difference in height. At this setting the system lies near instability, a single anomalous input of amplitude 20 as opposed the regular amplitude of 10 induces unstable behaviour despite the normal inputs that follow. For $\nu_1 = 4.2$ the regular input pattern induces exponential and unstable population explosion.

Small alterations to ν_2 cause drastic changes in behaviour. At 1.3 graceful system wide decay is induced. At 0.5 seemingly chaotic behaviour ensues, with many orders of magnitude separating peaks prompted by identical inputs.

Values of $\mu \geq 15$ incite system wide decay. For sufficiently small values, for example $\mu = 2.0$, reactions to inputs can be suppressed.

Under the standard system an input of $P = 4.0$ quickly induces system wide decay. However, when all the decay rates are halved regular responses are achieved. Likewise, responses to every input of $P = 1.0$ can be achieved by doubling all decay parameter values; under standard system parameters a response is mounted to alternate inputs only.

Parameters: $\tilde{\theta_1}$ and $\tilde{\theta_2}$
Effects: Reducing $\tilde{\theta_1}$ significantly, even to negative values, has little effect other than the speed with which the system settles; smaller values induce small initial proliferation peaks which take longer to stabilise. Large values cause large initial proliferation peaks, but the system always settles. Extremely large values induce erratic behaviour.

As with $\tilde{\theta_1}$, adjustments to $\tilde{\theta_2}$ cause initial instability. The heights of stable proliferation peaks can be slightly reduced with larger $\tilde{\theta_2}$. Very large values of $\tilde{\theta_2}$ can incite system wide decay.

These parameters yield no significantly exploitable effects. Adjustments can instil minor alterations to peak heights, and significant adjustments will induce either erratic behaviour, else system decay. They dictate how quickly cells start producing cytokines through manipulation of the input to the sigmoid function S. To have any significant effect on behaviour, the system should be engineered such that the outputs from S reside mainly between 0.0 and 1.0 rather than at those values themselves. Engineering such a system is difficult, and changes to any of the other parameters will quickly throw S's output back to the extremes.

Parameters: $\bar{\psi_1}$ and $\bar{\psi_2}$
Effects: Significant reduction of $\bar{\psi_1}$ produces erratic behaviour, with huge fluctuations in cell proliferation peak heights. Smaller reduction induces proliferation peaks of alternating heights. Larger values reduce peak heights, and may cause system wide decay if set excessively large.

Very small values of $\bar{\psi_2}$ prevent sufficient u_2 from entering the system, and system wide decay ensues. Large values can set the system off in an instability, which takes time to settle, and also cause alternating peak heights. Anomalous but regular inputs of increased amplitude have differing effects on a system exhibiting alternating peak heights. If the anomaly falls on what would be a tall peak, the shorter peaks are completely suppressed and the remaining tall peaks that follow experience amplification while the system settles into its previous behaviour. If the anomaly calls on a short peak, all peak heights average out initially, but return to their normal behaviour.

Maintaining equal, but larger values of $\bar{\psi_1}$ and $\bar{\psi_2}$ reduces the height of cellular proliferation peaks. Smaller values increase proliferation heights. These effects are attributed to the nonlinear influence of u_1.

Parameters: $W_{j\,k}$
Effects: The variables comprising the interaction matrix are highly complex, and system behaviour is collectively dependent on all of their values. They affect how much stimulus each cell receives regarding the secretion of a particular cytokine, which determines the rates at which cytokine concentrations can accelerate. These cytokine concentrations in turn affect the quantity of cells there are that produce them. External stimuli have no direct impact on secretion; the only variable terms in determining secretion rates are cytokine concentrations themselves. Thus the cytokine secretion rates are linked. It is easy to set up the interaction matrix in such a manner that system wide decay ensues, however unbounded explosion is impossible: explosion in one cytokine concentration will soon lead to explosion in the other.

In general, equal concentrations of the two cytokines will prevent cellular proliferation and can prompt system wide decay, because of u_1's nonlinear effect. For this reason, the initial concentration of u_1 is lower, and its decay rate higher. Setups in which u_2 accelerates more quickly (and from a higher concentration) than u_1 allows for bursts of cell proliferation before u_1 reaches a counteracting concentration. As the two acceleration rates close on each other the height of

cell concentration peaks will reduce. If u_1 acceleration is allowed to exceed that of u_2, system decay will usually follow.

Reducing u_1 secretion stimulation though setting values of $W_{1\,1}$ to large negative numbers (u_1 suppresses itself) increases the size of peaks. Excessive negative values causes unstable behaviours to emerge, since the accelerations of the two cytokines are so far out of step that their concentrations experience very large fluctuations. Positive values yield no alteration in behaviour.

Small values of $W_{1\,2} < 1$ provide taller peaks in cell proliferation; extend the maximum period for which the system can remain responsive; and make the system more responsive to inputs of shorter period, with discernable reactions to every input as opposed to every few. However, these proliferations can vary in size considerably and do not display a stable pattern, indicating a shift towards erratic behaviour. Larger values of $W_{1\,2} > 1$ are not as reactive for small periods as small values, but the resulting behaviours are more stable, otherwise no significant effect is apparent.

Small values of $W_{2\,1} < 0.4$ induce system wide decay, since initial thresholds for cytokine secretion are never sufficiently overcome; u_1 discourages its own secretion, and u_2 only receives stimulus from the presence of u_1 ($W_{2\,2} = 0$). Hence, if $W_{2\,1}$ is set too small, no bootstrapping occurs. Values of $W_{2\,1} > 1$ have no effect on the peak sizes, and do not significantly affect the maximum period of responsiveness.

Setting $W_{2\,2} < -0.5$ induces system wide decay, and $W_{2\,2} > 0.5$ has no significant effect. Between these values, increasing $W_{2\,2}$ induces larger proliferation peaks.

Scaling all $W_{j\,k}$ values equally can slightly alter the height of peaks. Scalars ≥ 1 converge towards relatively large but fixed peak heights. Small scalars affect peak heights in a nonlinear manner; factors smaller than 0.3 cause decay, and a factor of 0.5 induces smaller peaks than factors of 0.3 and 0.7 do.

Parameter: γ
Effect: γ controls the effect that a particular concentration of u_1 has on cellular proliferation. The system is highly sensitive to this parameter, very small adjustments can drastically alter behaviour.

Large values of γ allow relatively small concentrations of u_1 to strongly hamper proliferation, thus reducing peak heights. Reduced proliferation peaks generates fewer cytokines whose concentrations curtail sooner. Hence, larger γ reduces the maximum period to which the system remains responsive, and also renders the system more responsive to small periods, with each input inciting proliferation. When perturbed with a single anomalous but regular input of greater amplitude large γ will provide a dampening effect on the resultant proliferation.

For smaller γ, the opposite applies. Sufficiently small γ will induce erratic responses to equal and regular inputs.
Examples: $\gamma = 0.05$ maintains stable responses at a period of 3.5, whereas the original value of 0.1 quickly decays at this period. A value of 0.15 causes system wide decay for a period of 2.0, while smaller values remain responsive.

When perturbed with an anomalous input of double the amplitude (20 as as opposed to 10) larger values of γ provides a dampening effect, reducing oscillation in proliferation peaks to later stimuli. Small values of γ are more sensitive to such anomalies, requiring more time to resettle into a stable output pattern, and exhibiting greater fluctuations in doing so.

6 Alterations to Model

Based on the parameter investigations carried out above, observations are made about the system's behaviour and alterations are made to enhance its behaviours with respect to the criteria of Section 3.

6.1 Excessive Decay Is Detrimental

A prominent facet of the system is it's tendency to decay. If insufficient stimulus is provided over a period of time, cytokine and cell concentrations decay such levels that resurrection requires an abnormally strong input. From an engineering perspective this behaviour is undesirable. We wish to induce a response to a standard input regardless of the delay preceding it. In other words, we wish for the memory and behavioural dynamics provided by cytokines to have a bounded temporal reach beyond which a default behaviour will ensue.

The most prominent amendment that will correct this behaviour is the introduction of baseline concentration levels. Any decay that will reduce a concentration to below this predetermined level is disregarded. Baselines for all three variables are required: if either u_2 or v concentrations drop too low, cellular resurrection becomes impractical; if u_1 drops too low, then ensuring a default behaviour is impossible since proliferation would go unchecked.

We set baseline values and initial concentrations to the same values. This prevents the creation of additional parameters for implementation, and starts the system in 'default behaviour' rather than the frequent instabilities observed above. It also removes any requirement that an input be present at system startup to prevent decay.

6.2 Engineering Cytokine Memory

With the addition of baseline concentrations, we attempt to manipulate the effect of cytokine memory. Given an anomalous input arriving either early or late with respect to an otherwise regular pattern, the effect of cytokines might be to dampen or to amplify the system's reaction to this anomalous input. Examples of both these behaviours are demonstrated in Figures 2 and 3 respectively. Note that the period used here is 4.0, longer than the system without baselines could remain responsive towards. Following the regular input at time 16, an anomalous input occurs after a delay of 1.6 time units (as opposed to 4.0). After this anomaly regular inputs of period 4.0 continue.

For the specific timing anomaly described, both the dampening and amplifying behaviours could be induced. ν_1 was increased to 5.0 to provide amplification,

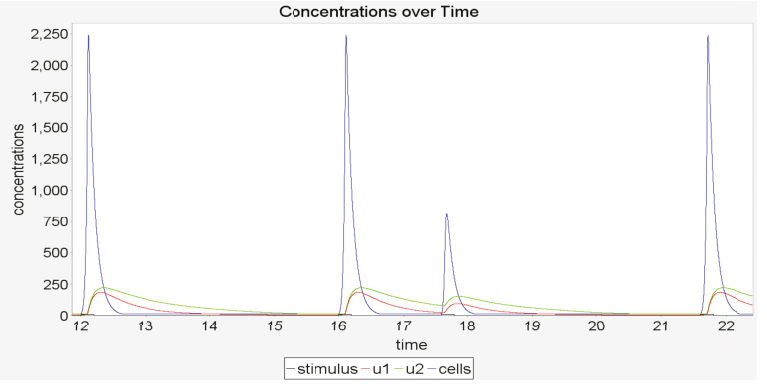


Fig. 2. Using concentration baselines to suppress reaction to early inputs. All parameter hold default values.

the standard system already provides dampening. When these settings are investigated in the context of other timing anomalies, the engineered behaviour is found to be inconstant. For the dampening system an anomalous delay of 1.0 induces no reaction. Increasing this delay causes a reaction of increasing magnitude, reaching a maximum for a delay of 2.4. Hereafter the system's reaction reduces in size again, reaching a low at a delay of 3.2. At 3.6 all variables have decayed to baseline values and default behaviour ensues.

For the amplification behaviour, at a delay of 0.4 or less no reaction is induced by the anomalous input. As the delay preceding the anomalous input increases past 0.6 an initially very small, but quickly increasing reaction is observed. By 0.8 the anomalous reaction peak magnitude exceeds that of the default behaviour. However, by 1.4 this peak starts to decrease in size, and by 2.8 is half the size of default behaviour peaks. Thereafter an increase to default behaviour ensues. Closer examination of this dip in peak size for a period of 2.4 reveals that when the anomalous input arrives concentrations of cells and u_1 have reached baseline levels, while u_2 resides at nearly 3 times its baseline level. One would expect this to result a strong proliferation, but it instead serves to hamper the reaction. The explanation is that the high levels of u_2 prompt speedy secretion of u_1, which prematurely stumps proliferation.

6.3 Realistic Decay Rates

The decay rates selected in the original paper, and utilised in the experiments above, are not biologically plausible. In vivo, cytokines decay at a significantly quicker rate than that at which cells die. Preliminary experiments with the ratios between cytokine and cell decay rates inverted yielded the following observations.

The temporal reach of cytokine memory is hugely reduced, and without baseline concentrations system wide decay is commonplace. Cellular proliferations exhibit very flat 'sawtooth' patterns. To avoid simply inciting 'default behaviour'

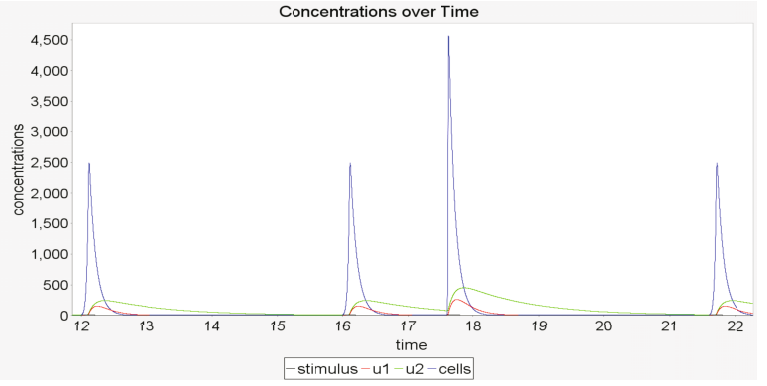


Fig. 3. Using concentration baselines to amplify reaction to early inputs. $\nu_1 = 5.0$.

input periods must be very short, at which point cell proliferation peaks and troughs are almost indistinguishable.

7 Suitability for Engineering

In this section we examine the system's suitability for use in an engineering context based on the experiments and alterations reported above and the criteria of Section 3.

Early experiments reported system wide decay in all variables was possible by 'starving' the system of sufficient input over time. This is generally undesirable. System wide decay could perhaps be harnessed in an anomaly detection setting, but one struggles to find additional value in employing this system over more conventional means. The introduction of baseline concentrations corrected the problem, providing a 'default behaviour' to late inputs, and ensuring reactivity to inputs of infinite delay. Conversely, reactivity to very short period inputs can be tailored as described above. Behavioural stability following anomalies in the input stream is possible, the extent to which is found to be largely dependent on the system's particular parameter values.

Regarding the requirement that system behaviour be malleable by adjustment of parameters, there does exist scope to affect system behaviour. However, there is a significant level of redundancy and pleiotropism in the parameters themselves. Adjustment of several parameters can bring about the same effect, for example, the height of proliferation peaks can be adjusted through independent manipulation of ν_1, ν_2, μ, $\bar{\psi}_1$, $\bar{\psi}_2$, $W_{1,1}$, $W_{1,2}$, $W_{2,2}$, or γ. However, adjustment to any one of these parameters will have cascade effects such as altering the maximum period for which the system remains responsive, the responsiveness to very low period inputs, or the time taken for the system to reestablish a stable behaviour following an anomalous input. Engineering an exact behaviour requires adjustment of several, if not all, parameters simultaneously. Care must be taken

when adjusting certain parameters, such as γ or $W_{2,1}$ as small adjustments have yielded considerable behavioural change.

The system's reaction to increasing inputs is initially good, with stable behaviour ensuing. However, the erratic reactions that follow from inputs of magnitudes > 50 are unwelcome. Deployment in an engineering field would require consideration regarding the magnitude of inputs generated.

Concentrations of cytokines and cells in the system residing above the baseline values exert a short term memory, and the value of deploying such a system for engineering lies in the ability to tailor the effect of this memory. However, in its current format the system is too challenging to accurately engineer, with an abundance of parameters which simultaneously affect several aspects of system behaviour. A reduction of the number of system parameters, and their cascade effects, through the replacement of finite predetermined constants with dynamically assessed ratios between system variable concentrations could potentially correct this problem.

8 Conclusions and Further Work

In this paper we introduced the work of Hone and van den Berg [2], and an empirical investigation of the dynamics exhibited by their 'numerical instantiation' was conducted. Analysis of the system's reaction to various patterns of input, and parameter adjustments was reported. Given these results, the addition of baseline variable concentrations was deemed necessary. The system's suitability for deployment in an engineering context was assessed, and was not deemed to be suitable in its current standing. A potential avenue for correction has been suggested. Spatial constraints have prevented us from displaying all our empirical evidence, these will be made available to interested parties on request.

The investigations carried out here are preliminary, and there is a great deal of scope for further work. This could include deliberately engineering into the system other properties of cytokines: pleiotropism; redundancy; antagonism; and synergy, which could prove useful for control systems which monitor multiple data sources. The addition of another cell type, and the system's dynamics concerning context switching between the two (interpretable through cell proliferations) presents an interesting path of investigation.

References

1. Hone, A., van den Berg, H.: In Silico Immunology. In: Mathematical Analysis of Artificial Immune System Dynamics and Performance, pp. 351–374. Springer, Heidelberg (2007)
2. Hone, A., van den Berg, H.: Modelling a cytokine network (special session: Foundations of artificial immune systems). In: FOCI, pp. 389–393. IEEE, Los Alamitos (2007)
3. Stepney, S., Smith, R.E., Timmis, J., Tyrrell, A.M., Neal, M.J., Hone, A.N.W.: Conceptual frameworks for artificial immune systems. International Journal of Unconventional Computing 1(3), 315–338 (2005)

An Empirical Study of Self/Non-self Discrimination in Binary Data with a Kernel Estimator

Thomas Stibor

Department of Computer Science
Darmstadt University of Technology
64289, Darmstadt, Germany
stibor@sec.informatik.tu-darmstadt.de

Abstract. Affinity functions play a major role within the artificial immune system (AIS) framework and crucially bias the performance of AIS algorithms. In the problem domain of self/non-self discrimination by means of negative selection, affinity functions such as the Hamming distance or the r-contiguous distance are frequently applied to measure distances in binary data. In recent years however, several limitations and problems with these distance measurements in negative selection have been identified. We propose to measure distances in binary data by means of probabilities which are modeled with a kernel estimator. Such a probabilistic model is preeminently applicable for the self/non-self discrimination problem. We underpin our proposal with an empirical study on artificially generated and real-world datasets.

1 Introduction

Self/non-self discrimination models are discussed intensively in immunology and also in the artificial immune system (AIS) community. In the field of AIS the negative selection is a popular, however also a controversial approach to discriminate self from non-self [1],[2]. The discrimination capability of negative selection is biased by the chosen shape space and the used affinity functions. In binary shape space (also called Hamming shape space) all immune components are represented as bit strings. The affinity between any two bit strings is measured with affinity functions such as the Hamming and r-contiguous distance. In recent years, however, research revealed that affinity functions used in negative selection induce manifold problems. The problems can be summarized as follows. Poor generalization capabilities, that is, the accurate self/non-self prediction of *unseen* bit strings [2]. Infeasible computational complexity of finding detectors [2]. To overcome these problems, it seems reasonable to look beyond the "classical" affinity functions proposed in the field of AIS.

The problem of self/non-self discrimination can be stated as follows. Given self data, that is, a sample $\mathcal{S}$ of bit strings which characterizes self:

– *Does an unseen bit string* $\mathbf{u}$ *belong to self?*

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 352–363, 2008.

This problem is usually tackled by using negative selection and corresponding affinity functions for binary data. Observe that this problem cannot be answered satisfyingly without giving a clear specification of self. In other words, the problem cannot be fitted in any machine learning framework.

By considering this problem from a statistical point of view, it can be equivalently formulated as follows:

– *Does* $\mathbf{u}$ *originate from the same probability distribution as bit strings in* $\mathcal{S}$*?*

This second question can be answered by assuming that $\mathcal{S}$ is i.i.d. generated by some unknown distribution which corresponds to self and that self data occurs concentrated. This leads to the problem of estimating the underlying probability distribution which generates $\mathcal{S}$ and finally to the rejection of data of low probability. Once the underlying probability distribution is properly modeled, membership queries, that is the first question, can be also answered.

In their seminal paper Kullback and Leibler stated [3]:

> *"We are also concerned with the statistical problem of discrimination by considering a measure of the "distance" or "divergence" between statistical populations in terms of our measure of information."*

By reviewing known problems in negative selection, it seems therefore reasonable to tackle the self/non-self discrimination problem by means of a statistical approach which will be discussed and empirically investigated in this paper. We structure the paper as follows: The kernel estimator method for binary data is explained in section 2. An experiment on artificially generated data is provided in section 2.1. The statistical discrimination function is presented in section 3. In section 4, an additional experiment is performed to explore whether regions where most of the self data is concentrated can be appropriately modeled. Results of detecting corrupted handwritten digits are presented in section 5. Conclusions and outlooks are provided in section 6.

2 Kernel Estimator for Binary Data

Kernel estimators belong to the class of non-parametric models and are well-known methods for estimating densities for continuous domains [4], [5]. For binary data, that is discrete data, kernel estimators such as Parzen Window or Nearest-Neighbor are not applicable due to their continuous nature. Aitchison and Aitken proposed a kernel estimator for binary data [6].

Given sample $\mathcal{S} = \{\mathbf{x}_t\}_{t=1}^{N}$ from $\{0,1\}^l$ and kernel function

$$K_h(\mathbf{x}|\mathbf{y}) = \begin{cases} h^{l-d(\mathbf{x},\mathbf{y})}(1-h)^{d(\mathbf{x},\mathbf{y})} & \text{for} \quad \frac{1}{2} \leq h < 1 \\ \begin{cases} 1 & (\mathbf{x} = \mathbf{y}) \\ 0 & (\mathbf{x} \neq \mathbf{y}) \end{cases} & \text{for} \quad h = 1 \end{cases} \tag{1}$$

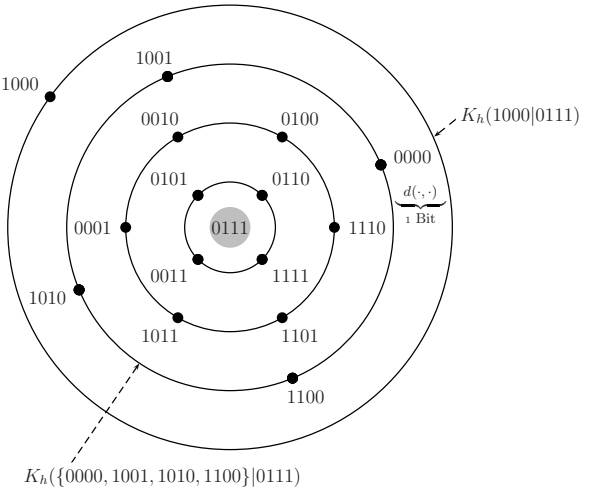


Fig. 1. Coherence between kernel function $K_h(\cdot|\cdot)$ and Hamming distance $d(\cdot,\cdot)$. The Hamming distance from 0111 to all bit strings sitting on the same ring is related to the probability mass function $K_h(\cdot|0111)$. Note that the Hamming distance is increasing from center 0111 to bit strings sitting on the outer rings at one bit per ring.

where

$$d(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \equiv \sum_{i=1}^{l} x_i \text{ XOR } y_i$$

is the Hamming distance, and h the bandwidth parameter. The true underlying probability distribution which corresponds to sample $\mathcal{S}$ can be estimated by:

$$\widehat{P}(\mathbf{x}|\mathcal{S}) = \frac{1}{N}\sum_{i=1}^{N} K_h(\mathbf{x}|\mathbf{x}_i). \tag{2}$$

The kernel function $K_h(\mathbf{x}|\mathbf{y})$ is a probability mass function and is related to the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ (see Fig 1). Loosely speaking, the smaller the Hamming distance the larger the probability. Analogous to continuous kernel estimators, the bandwidth parameter h in (2) controls the *smoothness*, i.e. the influence of the surrounding bit strings. The smallest bandwidth $h = 1/2$ gives the uniform distribution $\widehat{P}(\mathbf{x}|\mathcal{S}) = (1/2)^l$ for all $\mathbf{x} \in \{0,1\}^l$, whereas the largest bandwidth $h = 1$ gives the distribution of the relative frequencies.

To find an appropriate value of bandwidth parameter h such that consistency properties are obeyed, Aitchison and Aitken proposed to maximize:

$$\prod_{i=1}^{N} \widehat{P}(\mathbf{x}_i \,|\, \mathcal{S} \setminus \{\mathbf{x}_i\}) \tag{3}$$

where $\mathcal{S}\setminus\{\mathbf{x}_i\}$ denotes sample $\mathcal{S}$ with excluded bit string $\mathbf{x}_i$ (leave-one-out method).

Note that (3) can lead to numerical instabilities for large sample sizes. To avoid such a problem, one can also maximize the corresponding log-likelihood value:

$$\sum_{i=1}^{N} \log \widehat{P}(\mathbf{x}_i \,|\, \mathcal{S} \setminus \{\mathbf{x}_i\}). \tag{4}$$

It is worthwhile to notice that by maximizing (3), (4) respectively, one mutually minimizes the Kullback-Leibler divergence [3]:

$$\sum_{i=1}^{N} G(\mathbf{x}_i) \log \left(\frac{G(\mathbf{x}_i)}{\widehat{P}(\mathbf{x}_i|\mathcal{S})} \right). \tag{5}$$

The Kullback-Leibler divergence can be considered as a closeness measure between the true underlying probability distribution $G(\mathbf{x})$ and the estimated distribution $\widehat{P}(\mathbf{x}|\mathcal{S})$. The smaller the value of (5), the more "similar" are the true and estimated probability distribution.

2.1 Experiment on Data Generated by Mixture of Multivariate Bernoulli Distributions

For creating binary self data, it is helpful to use a *generative model* such that samples can be generated from the true underlying distribution which is specified by some parameters. A multivariate Bernoulli distribution is a generative model and fulfills this criterion. To be more precise, the distribution is specified by parameter vector $\boldsymbol{\Theta} \in [0,1]^l$ and takes binary values $x_i = 1$ with probability Θ_i and $x_i = 0$ with the complementary probability $1 - \Theta_i$, for $i = 1, \ldots, l$. It therefore has probability mass function:

$$P(\mathbf{x}|\boldsymbol{\Theta}) = \prod_{i=1}^{l} \Theta_i^{x_i} \, (1 - \Theta_i)^{1-x_i}. \tag{6}$$

To model higher order correlations in the generated samples, it is necessary to combine mixtures of multivariate Bernoulli distributions:

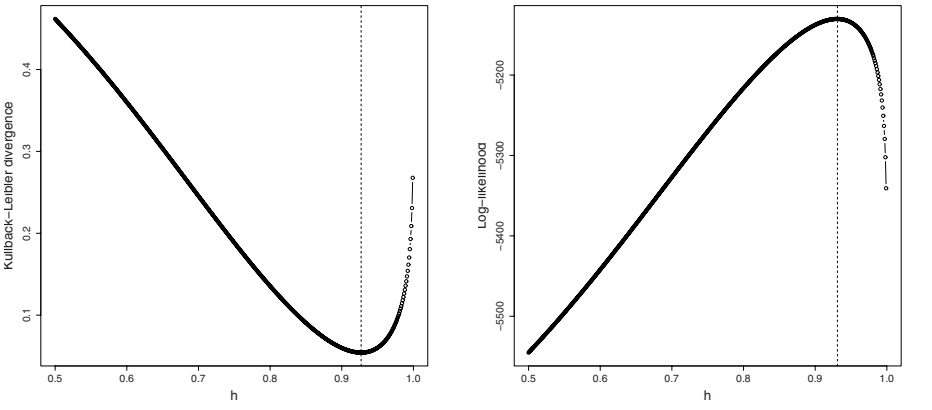
$$P(\mathbf{x}|\overline{\boldsymbol{\Theta}}, \boldsymbol{\alpha}) = \sum_{m=1}^{M} \alpha_m \, P(\mathbf{x}|\boldsymbol{\Theta}_m), \tag{7}$$

where the mixture proportion $\boldsymbol{\alpha} \in \mathbb{R}^M$ has to obey the convex combination $\sum_{m=1}^{M} \alpha_m = 1$ with $\alpha_m \geq 0$ and $\overline{\boldsymbol{\Theta}}$ is composed of $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_M)$.

In this experiment we specified $M = 3$ mixtures of multivariate Bernoulli distributions with following parameters:

$$\boldsymbol{\alpha} := \begin{bmatrix} \frac{1}{9} \\ \frac{3}{9} \\ \frac{5}{9} \end{bmatrix}, \qquad \overline{\boldsymbol{\Theta}} = \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \boldsymbol{\Theta}_2 \\ \boldsymbol{\Theta}_3 \end{bmatrix} := \begin{bmatrix} \frac{1}{10} & \frac{4}{5} & \frac{3}{5} & \frac{1}{5} & \frac{1}{5} & \frac{7}{10} & \frac{3}{10} & \frac{1}{10} \\ \frac{1}{2} & \frac{1}{5} & \frac{2}{5} & \frac{7}{10} & \frac{4}{5} & \frac{3}{5} & \frac{1}{10} & \frac{2}{5} \\ \frac{7}{10} & \frac{1}{10} & \frac{3}{10} & \frac{1}{5} & \frac{1}{2} & \frac{7}{10} & \frac{1}{2} & \frac{3}{5} \end{bmatrix},$$

and denote the true underlying distribution as $G(\mathbf{x}) \equiv P(\mathbf{x}|\overline{\boldsymbol{\Theta}}, \boldsymbol{\alpha})$.



(a) The dotted line denotes the value of h where the smallest Kullback-Leibler divergence value between the true probability distribution $G(\mathbf{x})$ and kernel estimated probability distribution $\widehat{P}(\mathbf{x}|\mathcal{S})$ can be found.

(b) The dotted line denotes the value of h where the largest log-likelihood value of $\widehat{P}(\mathbf{x}|\mathcal{S})$ can be found.

Fig. 2. Coherence between kernel parameter h and Kullback-Leibler divergence (left), and log-likelihood evaluation by means of the leave-one-out method (right). The value of h which maximizes (4) corresponds to the smallest Kullback-Leibler divergence value.

In non-parametric models *no* parametrized distribution has to be fitted in the samples; therefore, one has to determine only the suitable bandwidth parameter h. In this experiment the parameter h is run from 1/2 to 1. The corresponding value of (4) as well as the Kullback-Leibler divergence between $G(\mathbf{x})$ and $\widehat{P}(\mathbf{x}|\mathcal{S})$ are depicted in Figure 2.

One can observe that by maximizing (4) one mutually minimizes the Kullback-Leibler divergence between true the probability distribution and the kernel estimated. To say it the other way around, given a sample $\mathcal{S}$ which characterizes self and bandwidth parameter h which maximizes (4). One can model the underlying probability distribution which corresponds to $\mathcal{S}$ and hence is able to discriminate self from non-self by means of probabilities. Note that the Hamming distance is still used as a measurement, however expressed in terms of weighted kernel estimated probabilities. This allows the modeling of smooth discrimination boundaries, whereas the plain Hamming distance does not offer such degrees of smoothness (see [7]).

3 Statistical Discrimination in Binary Data

Let $\mathcal{S}$ be a sample which characterizes self and h the bandwidth parameter which is found such that (4) is maximized. A probabilistic discrimination function for the self/non-self problem[1] can be defined as follows:

[1] In the field of machine learning this equivalent problem is termed outlier detection or novelty detection.

$$\mathfrak{D}(\mathbf{x}, t) = \begin{cases} \widehat{P}(\mathbf{x}|\mathcal{S}) \geq t, & \text{self} \\ \text{otherwise}, & \text{non-self} \end{cases} \tag{8}$$

where $\mathbf{x}$ is the to classified bit strings and t some threshold. By specifying a value for t, one obtains enclosed decision region(s) such that most of the support of the distribution is captured. In other words, if $\mathbf{x}$ is within the region(s) where most of the self data is concentrated, then $\mathbf{x}$ belongs to self otherwise it belongs to non-self. It is worthwhile to mention that discrimination function $\mathfrak{D}$ can be extended to a multi-class decision function by assigning $\mathbf{x}$ to that class where the corresponding class-conditional probability is largest.

4 Experiment on Data Generated by Mixture of Gaussian Distributions

Due to the fact that mixtures of multivariate Bernoulli distributions are hardly to visualize, a second experiment is performed. In this experiment we explore whether regions, where most of the self data is concentrated, can be appropriately modeled. Therefore, self data is generated by a mixture of 2-dim. Gaussian distributions with different mean vectors and covariance matrices and consists of 5000 data points. The generated self data is visualized in Figure 3(a), the corresponding density image is depicted in Figure 3(b).

One can see in Figure 3(a) that self data is concentrated in regions of high density. This coincidence with our assumption and leads to the problem of finding regions where most of the self data is concentrated.

Note that the domain of (2) is $\{0,1\}^l$. We therefore use the mapping from $\mathbb{R}^2 \rightarrow \{0,1\}^l$ proposed in [8]. That is, the data is min-max normalization to $[0,1]^2$ and discretized to bit strings of length $l = 16$

$$\underbrace{b_1, b_2, \ldots, b_8}_{b_x}, \underbrace{b_9, b_{10}, \ldots, b_{16}}_{b_y},$$

where the first 8 bits encode the integer x-value

$$i_x := \lceil 255 \cdot x + 0.5 \rceil$$

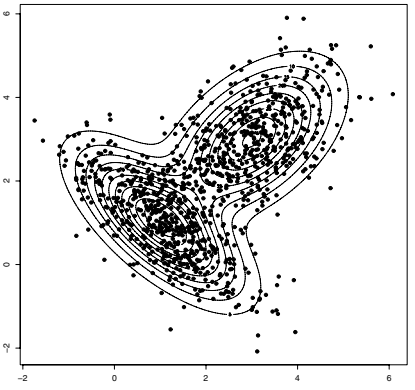
and the last 8 bits the integer y-value

$$i_y := \lceil 255 \cdot y + 0.5 \rceil,$$

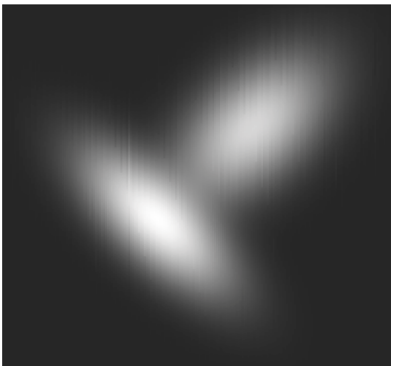
that is,

$$\begin{aligned} [0,1]^2 &\rightarrow (i_x, i_y) \in (1, \ldots, 256) \times (1, \ldots, 256) \\ &\rightarrow (b_x, b_y) \in \{0,1\}^8 \times \{0,1\}^8. \end{aligned}$$

By means of the leave-one-out method bandwidth parameter $h = 0.909$ is determined. The corresponding density image is depicted in Figure 4(b), where each

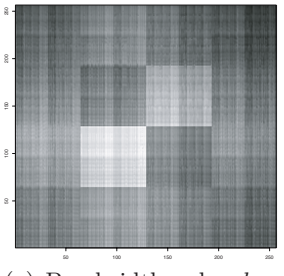


(a) Self data is generated by a mixture of two multivariate Gaussian distributions with different mean vectors and covariance matrices.

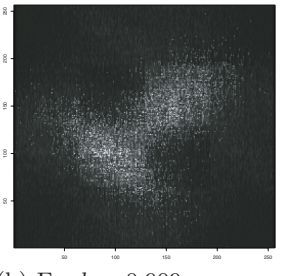


(b) Density image of the underlying distributions. Self data is concentrated in regions of high probability (light regions).

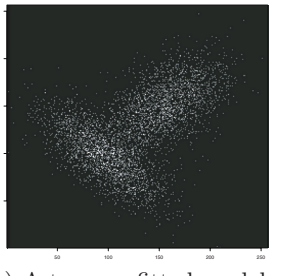
Fig. 3. Self data is sampled from a mixture of multivariate Gaussian distributions



(a) Bandwidth value $h = 0.55$ results in a too underfitted model.



(b) For $h = 0.909$ a proper generalization is obtained.



(c) A too overfitted model $h = 1$ results in a poor generalization.

Fig. 4. Coherence between different bandwidth values and estimated models

pixel in the 256×256 grid represents a bit string of length $l = 16$. The color corresponds to the probability $\widehat{P}(\mathbf{x}|\mathcal{S})$. For the sake of comparison, two additional density images of bandwidth value $h = 0.55$ and $h = 1$ are depicted (see Fig. 4(a), 4(c)). One can observe that the true underlying distribution can be closely approximated if an appropriate value of h is determined. For a too oversmoothed bandwidth value $h = 0.55$ the resulting model is underfitted, whereas for $h = 1$ the model is overfitted. For $h = 0.909$ the probability distribution is appropriately modeled, thus good generalization is obtained.

5 Handwritten Digit Recognition Experiment

Recognizing handwritten digits is a challenging real-world problem in the field of machine learning. In this experiment, we focus on the problem of outlier detection, that is, the detection of digits which are corrupted. In the language of self/non-self discrimination, self of each digit is modeled as shown in section 2 and corrupted digits are detected by means of decision function (8).

To obtain meaningful results regarding the robustness of the kernel estimator method, experiments on two popular handwritten digits datasets (USPS and MNIST database) are performed.

The USPS database[2] contains handwritten digits scanned from envelopes by the U.S. Postal Service. The digits are size-normalized in a 16×16 fixed image of gray color values in the range $[-1, 1]$. The database consists of 7291 training examples and 2007 testing samples which are partitioned in digit sets 0 to 9 (see Table 1).

Table 1. Number of digits in training and testing set in the USPS database

digit	0	1	2	3	4	5	6	7	8	9
training set	1194	1005	731	658	652	556	664	645	542	644
testing set	359	264	198	166	200	160	170	147	166	177

The USPS database contains a number of corrupted digits, which not even humans can correctly classify (human error rate 2.5%) and therefore is a challenging benchmark. However, the database is also criticized due to their noisy nature [9].

The MNIST database[3] contains also handwritten digits. However if one compares the two databases, then one can observe that the MNIST database has cleaner digits thus becomes the state of the art benchmark database in recent years. The digits in the MNIST database are centered and size-normalized in a 28×28 fixed-size image of gray color values $\{0, 1, \ldots, 255\}^{784}$. The MNIST database consists of 60000 training examples and 10000 testing samples which are partitioned in digit sets 0 to 9 (see Table 2).

Table 2. Number of digits in training and testing set in the MNIST database

digit	0	1	2	3	4	5	6	7	8	9
training set	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949
testing set	980	1135	1032	1010	982	892	958	1028	974	1009

To obtain comparative results between the two databases, digits in the USPS database are min-max normalized from $[-1, 1]$ to gray color values $\{0, 1, \ldots, 255\}^{256}$. Both databases are finally binarized by means of:

[2] Available at: `http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/`
[3] Available at: `http://yann.lecun.com/exdb/mnist/index.html`

$$\mathfrak{B}(\mathbf{z}, t_{\mathtt{bw}}) = \begin{cases} z_i \leq t_{\mathtt{bw}}\,, & 0 \\ \text{otherwise}\,, & 1 \end{cases} \tag{9}$$

where threshold $t_{\mathtt{bw}} = 128$ is chosen and $\mathbf{z} \in \{0, 1, \ldots, 255\}^{256}$ (USPS database), $\mathbf{z} \in \{0, 1, \ldots, 255\}^{784}$ (MNIST database), respectively.

The bandwidth value h of each digit class for both training sets is determined by means of the leave-one-out method and results in:

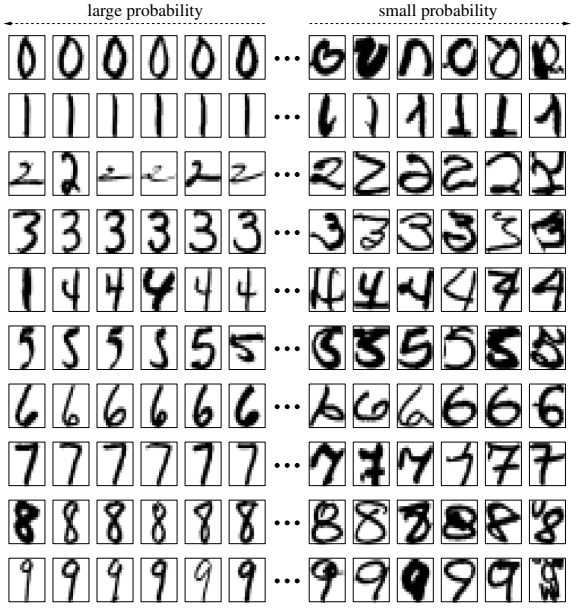
	digit	0	1	2	3	4	5	6	7	8	9
USPS	h	0.917	0.99	0.871	0.888	0.906	0.877	0.92	0.938	0.889	0.93
MNIST	h	0.94	0.984	0.928	0.935	0.945	0.936	0.946	0.956	0.929	0.95

5.1 Results

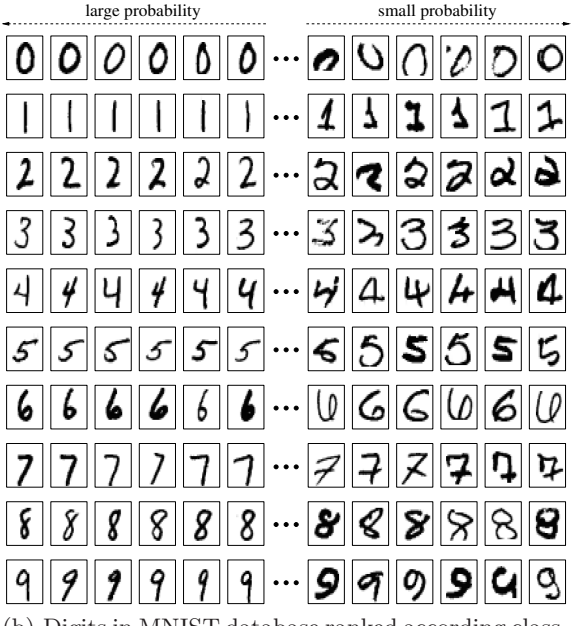
Both testing sets contain no information regarding the magnitude of corruption of the digits. As a result, it is difficult to obtain meaningful outlier detection results. Due to such difficulties, the digits of each class are ranked. To be more precise, the digits of each class are ranked in descending order regarding their class-conditional probabilities (see Fig. 5). One can see that corrupted digits have small class-conditional probabilities and hence can be recognized as outliers by decision function $\mathfrak{D}$ with regard to some threshold value t. Furthermore, one can observe that some less corrupted digits ("7") which are written according to the European standard have small probabilities. This is an undesirable result and is caused by the fact that the training set contains an underrepresented amount of those digits. This problem can be addressed by tuning the corresponding bandwidth parameter towards more smoothness. Moreover, one can observe that in the USPS database the mislabeled digit 1 has a large estimated probability and thus can not be detected as an outlier.

Table 3. State of the art classification results on testing sets USPS and MNIST. For a detailed overview see [9], pp. 219 and pp. 341.

Database	Classifier	Error rate (%)
USPS	Linear SVM	8.9
	Relevance Vector Machine	5.1
	Hard margin SVM	4.6
	SVM	4.0
	Hyperplane on KPCA features	4.0
	Kernel Fisher Discriminant	3.7
	Virtual SVM	3.2
	Virtual SVM, local kernel	3.0
MNIST	Linear classifier	8.4
	3-Nearest-Neighbor	2.4
	⋮	⋮
	Virtual SVM with 8 VSVs per SV	0.6
	Virtual SVM with 12 VSVs per SV	0.6



(a) Digits in USPS database ranked according class-conditional probabilities in descending order.



(b) Digits in MNIST database ranked according class-conditional probabilities in descending order.

Fig. 5. First six digits of each class (testing set) ranked according to the largest, smallest class-conditional probability, respectively. One can see that corrupted digits have smaller probabilities compared to "clean" digits having larger probabilities.

In terms of the overall classification error rate[4], the following results are obtained on the testing sets: USPS database 7.47 % and MNIST database 3.92 %. Compared to the state of the art classification results (see Table 3) our achieved results are limited competitive. However, one has to mention that the best achieved classification results are obtained with highly tuned classifiers which are invariant regarding translation and rotation. Furthermore, we used binary features rather than gray color values from $\{0, 1, \ldots, 255\}$ and therefore utilized a poorer feature representation due to the operation on binary data. On the other hand one should mention that kernel based estimation methods suffer of high computational complexity. This results from the fact that each bit string is used to evaluate term (2). However there exist different techniques for reducing the computational complexity of kernel based estimation methods (e.g. [10],[11]). These techniques can be also applied to reduce the computational complexity of term (2). Additional improvements regarding the detection accuracy could be obtained by applying different binarization techniques.

6 Conclusion

Self/non-self discrimination in binary data is a challenging problem in the field of AIS. It has been tackled with negative selection and affinity functions such as the Hamming and the r-contiguous distance. Research results in recent years, however, revealed manifold problems in negative selection with regard to the generalization capability, and with regard to the computational complexity. We proposed to model *self* by means of a statistical approach, namely by estimating the underlying probability distribution which corresponds to self with a kernel estimator. The proposed method was motivated by the fact that the self/non-self discrimination problem can be clearly specified from a statistical view point. Such a statistical method is far from any immune-inspired paradigms, however, overcomes known problems in the immune-inspired negative selection method. From our point of view it is worthwhile to introduce such a statistically founded method in the field of AIS. It allows us to consider problems formulated in the field of AIS from a mathematically founded perspective, rather than by biologically motivated arguments. Observe that in the early days the term "neural network" was motivated towards modelling networks of real neurons in the brain. Nowadays:

> *"The perspective of statistical pattern recognition, however, offers a much more direct and principled route to many of the same concepts."* [Neural Networks for Pattern Recognition, C. M. Bishop]

Acknowledgment

The author thanks Erin Gardner for her valuable suggestions and comments.

[4] Recall: discrimination function $\mathfrak{D}$ can be extended to a multi-class decision function by assigning $\mathbf{x}$ to that class where the corresponding class-conditional probability is largest.

References

1. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proceedings of the Symposium on Research in Security and Privacy, pp. 202–212. IEEE Computer Society Press, Los Alamitos (1994)
2. Stibor, T.: On the Appropriateness of Negative Selection for Anomaly Detection and Network Intrusion Detection. PhD thesis, Darmstadt University of Technology (2006)
3. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
4. Duda, R., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Chichester (2001)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
6. Aitchison, J., Aitken, C.G.G.: Multivariate binary discrimination by the kernel method. Biometrika 63(3), 413–420 (1976)
7. Stibor, T.: Discriminating self from non-self with finite mixtures of multivariate bernoulli distributions. In: Proceedings of Genetic and Evolutionary Computation Conference – GECCO. ACM Press, New York (to appear, 2008)
8. González, F., Dasgupta, D., Gómez, J.: The effect of binary matching rules in negative selection. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 195–206. Springer, Heidelberg (2003)
9. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
10. Giromali, M., He, C.: Probability density estimation from optimally condensed data samples. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10), 1253–1264 (2003)
11. Fukunaga, K., Hayes, R.R.: The reduced parzen window classifier. IEEE Transaction on Pattern Analysis and Machine Intelligence 11(4), 423–425 (1989)

The Pathways of Complement

Jonathan M. Aitken, Tim Clarke, and Jonathan I. Timmis

Department of Electronics,
University of York, Heslington, York, YO10 5DD, UK
jma105@ohm.york.ac.uk, tc@ohm.york.ac.uk, jt517@ohm.york.ac.uk

Abstract. The natural world has developed very effective methods for dealing with pathogens that are invading an organism. By taking inspiration from the in-built, innate, response this paper develops a new algorithm that mimics activation path of the Alternative Pathway of Complement. The Alternative Pathway is triggered by cell surfaces. If the surfaces display safe characteristics then a strong suppression process prevents activation. In the case of non-safe surfaces a positive feedback loop rapidly identifies the location marking it for removal by phagocytosis.

1 Introduction

This position paper is about solving a difficult engineering problem using techniques inspired by biological systems, viewed at both the organism population level and the organism cellular level.

The problem itself is generic. A collection of fallible engineering entities is required to interact to achieve some behavioural goals. Through mutual observation of their behaviours, they must be able to recognise a failure, locate the one of more failing entities and subsequently respond to re-establish goal directed, normal network behaviour. Typical target system could include swarms of Unmanned Aerial Vehicles (UAVs) operating co-operatively to render a mission successful, ad-hoc heterogeneous communications networks which respond to loading variations or node failures [1] and any distributed control system which relies on may rely on a combination of subsystems and sensors such as power distribution networks [2].

1.1 Socially Attentive Monitoring

The recognition of failure through mutual observation has been considered in a social population context under the guise of Socially Attentive Monitoring. The concept of Socially Attentive Monitoring uses observed aspects of human social behaviour as a driving force for identifying the health of a system. Socially Attentive Monitoring was first investigated by social scientists. Festinger [3] proposed three hypothesis about how humans react in a social situation. The first hypotheses states that "there exists, in the human organism, a drive to evaluate his opinions and his abilities" - therefore any human carrying out a task wishes to work out how well they conduct that task.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 364–375, 2008.

Festinger's second hypothesis states that "to the extent that nonsocial means are available, people evaluate their opinions and abilities by comparison respectively with the opinions and abilities of others". Therefore humans seek out those that they see as having abilities or opinions in the same area and then judge themselves accordingly.

The third hypothesis acts to limit this desire for comparison stating that "the tendency to compare oneself with some other specific person decreases as the difference between his opinions or ability and one's own increases". Therefore humans only compare themselves to those that they see of a similar ability or opinion, therefore effectively searching out peers for accurate comparison.

Festinger's work has been picked up by the Distributed Artificial Intelligence community and used in several different applications. Kaminka and Tambe [4,5,6] have investigated the use of Socially Attentive Monitoring in the concept of agent coordination.

In control engineering a key property of a system are the natural modes. A collection of continuous, complex exponential values that determine how a system responds to a given input. This work uses the identified modes of each network component as a *signature* representing how that element is behaving at that moment in time. From this point the immune inspired algorithm is used to identify the affected element.

The remainder of this paper focusses on extending the idea of signatures using inspiration from the complement system to identify point(s) of failure. Subsequent action will be application specific, but we develop a simple mechanism to demonstrate the principles.

The paper is organised as follows. We begin by describing the current views of the three biological complement pathways as a focus for inspiration. In Section 2 the Alternative Pathway is introduced; from this an Algorithm is developed in Section 3 and compared to the equivalent biological route. Finally method for comparing the natural modes is outlined in Section 4.

2 The Complement Process

The complement system is named because it was believed to aid the antibody recognition process. However, it is an important system in its own right with a central role in the inflammatory response of the immune system [7,8,9], being sensitive to small amounts of pathogen. However, it has been found to be effected earlier in the process of the immune response than during any antibody action. The complement process is a combination of many different chemicals that produce a wide range of effects across the immune process.

There are three top level routes into the complement process, namely the Classical Pathway, the Mannose-Binding Lectin (MBL) Pathway and the Alternative Pathway. Each of these pathways uses a variety of different chemicals during the process, but at the centre of each process is the C3 Convertase. The basic complement components are labelled C1 to C9, although, in addition, the MBL Pathway uses mannose-binding-lectin-associated serine proteases (MASP-1 and

MASP-2) and MBL. The Alternative Pathway uses another group of chemicals whose shorthand names are denoted by capital letters. In the common notation, when a Cx element is cleaved it is split into two components denoted Cxa and Cxb, where x represents the number. The Cxb component is the larger fragment.

2.1 Classical Pathway Trigger

The Classical pathway is started by the C1 complement component. C1 is a complex of C1q and the inactive enzymes C1r and C1s. The Classical pathway starts when the C1q component either binds to antigen or the surface of a pathogen [10]. When the C1q head binds, it activates C1r which in turn, activates C1s. Activated C1s is capable of cleaving both C2 and C4 [11]. Firstly C4 is cleaved into C4a and C4b. C4b may attach to the pathogen surface. C4b binds C2 which can then be cleaved by C1s to form the C3 convertase C4b2b which remains bound to the pathogen surface [12].

2.2 Mannose-Binding Lectin Pathway Trigger

The MBL Pathway is initiated by pathogens containing Mannose and similar sugars [13] on their surface. Mannose is a good indicator of pathogen as, in vertebrates, the substance is typically shielded by sialic acid [12]. Although MBL is similar in structure and lineage [14] to the C1 protein [15], experimentation has shown it to be different, operating with less specificity [16].

Upon binding to a pathogen surface Mannose Associated Serine Protease (MASP)-2 becomes activated this cleaves C4 with the C4b element remaining attached. C2 binds to the C4b and is readily cleaved to C2b by MASP-2 forming the C3 convertase C4b2b [12,17].

2.3 Alternative Pathway Trigger

The Alternative Pathway of complement differs significantly from the other routes as it leads to a different C3 convertase [12]. The Alternative Pathway also differs as it is not triggered by just a pathogen, it is activated by the presence of any surface with regulation to ensure that the host is not harmed. In addition it possesses a different set of proteins that trigger the process, namely Factors B, D, H, I and Properdin. Figure 1 shows the process of the Alternative Pathway. At the heart of the Alternative Pathway of complement is the natural and spontaneous process of the conversion of C3 to C3(H_2O) [20,21] in fluid, a functional equivalent version of its cleaved component C3b in binding to Factor B although similar in structure to C3 [19]. The hydrolysed C3(H_2O) binds to Factor B in the presence of Magnesium ions forming C3(H_2O)B, whereupon Factor B in this compound is cleaved by Factor D to form the short-lived fluid phase C3 Convertase C3(H_2O)Bb [19].

The fluid phase C3 convertase cleaves local C3, the C3a produced diffuses away from the site. C3b attaches to a cell surface. This process must be completed quickly as C3b is rapidly inactivated [12]. The C3b on the surface binds to Factor

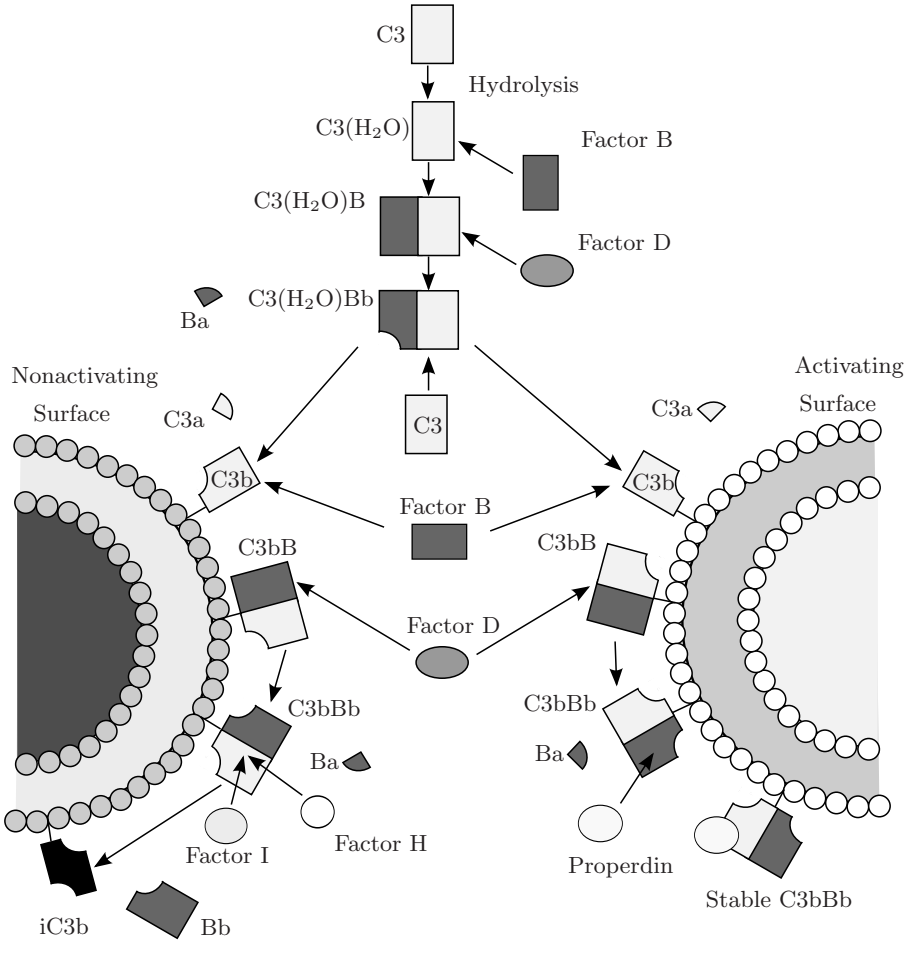


Fig. 1. Alternative Pathway of Complement (Adapted from [18] using [12] and [19])

B which is acted on again by Factor D to form the C3 convertase C3bBb attached to a cell surface.

Up to this point in the process there has been no differentiation between the surface of an invading pathogen and the host. It is the nature of the surface that determines whether complement is activated. This recognition is controlled through the β1H globulin known as Factor H [22,23]. In host cells Factor H preferentially binds to C3 convertase on the surface of a cell [24] which promotes inactivation by Factor I [25]. This inactive form, iC3b, is further acted upon by Factor I to produce C3c and C3dg [21] that can then be removed.

If Factor H does not act on the C3 convertase, properdin, acts to stabilise the compound. This reduces any potential effect of Factor H [26] providing it with a significantly longer half-life [27].

2.4 C3 Convertase to the Membrane Attack Complex

The result of the triggering of all three pathways is the production of C3 convertase which is capable of cleaving more C3 which then acts to re-enforce the Alternative Pathway providing a strong amplification loop via positive-feedback [27]. Any C3 cleaved by the C3 Convertase covalently attaches to the surface leaving the pathogen coated in C3b.

The next step in the process is the formation of the C5 convertase which results from the binding of an additional C3b molecule to the C3 convertase.

The C5 convertase cleaves C5 into C5a and C5b. The C5b molecules attach to the C5 convertase whereas the smaller C5a migrate away from the site of the pathogen. The small components of complement (C3a, C4a and C5a) are known as anaphylatoxins, as they can cause inflammation at the site. These small components diffuse away from the site of infection. Anaphylatoxins are also chemotaxins: phagocytes follow the diffusion gradient and so are attracted to the infection site. Effectively phagocytes are guided to the infection site by the distribution of the smaller complement components. The C3b attached to the surface of the pathogen acts as an opsonin, promoting phagocytosis of the pathogen by the phagocytes that have been recruited to the site by the anaphylatoxins.

The remainder of the complement components are involved in the terminal phase of the process. C6, C7 and C8 successively bind to C5b bound to the C5 convertase to produce a compound which results in the formation of a C9 polymer which crosses the membrane causing cell lysis [12].

2.5 Factor H

Factor H [28] is crucial in the identification of danger within the Alternative pathway of complement. It exists in high concentration in the blood plasma of humans [29] and its absence or mutation can result in serious harm to the host [30]. It acts to start the process of inactivation of C3 convertase only if found on host surfaces. Once this has been achieved it promotes the binding of Factor I to complete the inactivation.

Several experiments have been conducted examining the effect of Factor H on C3 Convertase bound to surfaces. The first molecule reported to influence C3b deposition on a surface was Sialic Acid [31]. Further investigation revealed that other polyanionic molecules such as the glycosaminoglycans and sulphated polysaccharides [29], of which heparin is a notable case - being a strong polyanion.

This binding of polyanions and sialic acid is especially significant as it provides host cells with a method for protecting themselves from Alternative Pathway complement attack. The presence of Factor H gives this key ability to differentiate host and pathogen surfaces [32]. Therefore any surface expressing polyanionic molecules will be deemed as not being dangerous to the host and so inhibit complement activation. The inability of molecules to express these key compounds can result in unchecked action of the complement cascade resulting in damage to the host [33].

The host possesses a method for suppressing complement activation via the Alternative Pathway by expressing polyanionic molecules. However, pathogens

have developed methods for using such tactics to avoid detection [34] by, for example, binding sialic acid or Factor H to their own surfaces [35] in order to mask themselves from the Alternative Pathway. Therefore the Alternative Pathway is capable of protecting the host unless the invading pathogens become "host-like" [36].

3 Applying Ideas from the Innate Immune System

The review of the Innate Immune System has revealed a collection of complex processes that co-operate together to provide not only an accurate and reliable indication of pathogen infiltration but are also capable of directing the initial stages of the immune attack to the infected area. Therefore the Innate Immune System provides a system worthy of investigation in its ability to protect a group of individual units. In order to protect such a group it would be beneficial to utilise joint information between the separate units. This scheme is typical of Socially Attentive Monitoring (SAM) [3,4]. By sharing information, the members of the group are then capable of identifying failures within their network. To this end, we have devised the Alternative Pathway Algorithm.

3.1 The Dynamic Element Swarm Problem

The problem considered in this paper consists of a fallible swarm network. A set of dynamic swarm elements must navigate a simple slalom course as a coherent group within a time limit. Should any element change dynamic behaviour characteristics such that the swarm interactive behaviour impedes the collective group temporal goal achievement, it must be identified and corrective action taken.

3.2 The Alternative Pathway Algorithm

This algorithm is based around the interactions at surfaces during the Alternative Pathway of complement and the utilises several key features:

- The Alterative Pathway relies on having large numbers of molecules of the element C3 circulating in the blood stream that are then allowed to interact with any surface that they meet. The process is widely distributed. Not all of the possible sites are compared at once - and there is no need to compare one site with every other in order to make a local decision, since knowledge about safe factors is widespread.
- The interaction with surfaces is non-deterministic but is influenced by the state of the surface at that time. If the surface is in the process of being identified as harmful, it is more likely that deposition will occur.
- The process of deposition is modified by the presence of expressible polyanionic molecules by the surface. These can prevent further deposition, causing inactivation and removal of the initial C3 from the bloodstream.
- The process of expression of the molecules is dependent on the nature of the surface. Pathogens may acquire Factor H or sialic acid which masks their true nature and prevents further activation.

– Failure to inactivate the deposition results in rapid, exponential growth through positive feedback which alerts the body to the problem location, attracting more deposition of C3b and attracting neutrophils as a result of release of chemoattractants from the process.

The implementation of the algorithm focuses on the process of message passing between software agents. The key features of the algorithm can be directly compared with the process of the Alterative Pathway:

– The elements are widely dispersed but have natural paths of communication between each other. There is no need for specific comparison in such a group. An element need not compare itself to the entire group at once. Therefore the comparison can be allowed to circulate information around the group. However, attention needs to be directed towards problem areas in a similar method to the release of chemoattractants.
– The process relies on the ability of each agent to identify a picture of how it is behaving at that instant in time. These signatures can be viewed as encapsulating the behaviour style of the agent for that period of time. In essence, their behaviour is similar to C3, which is allowed to flow freely around the network randomly, selecting a surface.
– These signatures can be used to modulate the behaviour of the network. A mismatch between a local signature and one received from another agent implies that all is not correct. Therefore, a matching function is required which can measure the separation between two signatures. If they are close, the process is similar to polyanionic expression, allowing inactivation. As the distance increases, the desire to compare with neighbours must be increased, as there is the potential that a problem has occurred but further investigation is required. This is analogous to the stabilisation of C3 convertase on the surface by properdin with additional release of chemoattractants.

Algorithm 1, reflects these basic properties found in the Alternative Pathway. The algorithm is implemented on each agent separately, within the context of an agent framework such as the Java Agent Development Framework [37]. The basic terms used in the algorithm are outlined below:

– $sig_{self}(t)$ - The signature of the agent running the algorithm at time t representing the surface of the agent.
– $sig_{peer}(t-x)$ - The signature of another agent in the system at a time x units prior to the present time. This represents circulating components of C3, free to move in the blood stream, eventually coming into contact with a surface.
– $C3_{self}$ - The stress of the agent at that time through complement deposition. This is linked to how likely an agent will accept a signature for comparison from a peer in the network. The lower the value the more likely a comparison is to take place. Therefore its level acts as an attractant for signatures circulating in the network, like an anaphylatoxin. The variable gives an indication of the level of stabilised C3 Convertase deposition on the surface.

- $match()$ - This function acts to generate a score indicating how similar the signature from a peer is to the agent at that time.
- R - A value representing the result of the matching function. The nearer to unity the value then the more similar the two pairs. Therefore, it acts as a stimulant for polyanionic compounds. At values close to unity, it will be more likely that the signatures match and that the surfaces are safe.
- $C3_{inc}$ - If signatures match $C3_{self}$ is incremented by this constant value. This results in suppressed activation of the cascade following a successful surface match.
- $C3_{max}$ - Maximum value of agent stress, representing the minimum level of C3 deposition on the cell. In order to maintain some comparison, the level must be set to less than unity so that there is some probability of comparison.
- $C3_{dec}$ - The decrement equivalent of $C3_{inc}$. It promotes activation of the cascade on an unsuccessful surface match.
- T_{MaxSig} - Maximum age of a signature. Any signature older than this limit is removed from the system.

4 The Matching Problem

It is essential that signatures from other peers are correctly compared at each agent location. Therefore a matching function is required that can successfully cope with a variable length list of complex valued numbers. The basis for the matching function is the Cauchy-Schwarz inequality [38] (1).

$$| < a, b > | \leq \|a\|_2 \|b\|_2 \tag{1}$$

Where $< . >$ denotes the inner product and $\|.\|_2$ denotes a 2-norm. Therefore any evaluation of the function in the form of (2) will provide a result that lies between zero (no similarity) and unity (identical).

$$R_1 = \frac{| < a, b > |}{\|a\|_2 \|b\|_2} \tag{2}$$

Selecting a random set of six eigenvalues of a discrete state space system to be 0.9969, 0.9698, $0.9768 + 0.1456i$, $0.9768 - 0.1456i$, 0.9434 and 0.9234. The effectiveness of (2) can be evaluated by moving the pair of complex poles around the available space, the unit circle of the argand diagram achieving the result in figure 2(a). A good match of eigenvalues can be achieved through (2) when the two vectors are closely aligned. As the eigenvalues drift apart then the value of R_1 decreases but for very dissimilar vectors then there is insufficient distinction.

Additionally (2) does not give any distinction between a doubling of eigenvalues. The equation is normalised so that then providing that the shape of the values contained in the vector are identical the result will be the same. Therefore (2) can be modified to be adjusted to be normalised to the maximum norm of the two vector of eigenvalues so that a doubling in the total eigenvalue will produce a lower match.

Algorithm 1. Algorithm to Mimic the Alternative Pathway of Complement

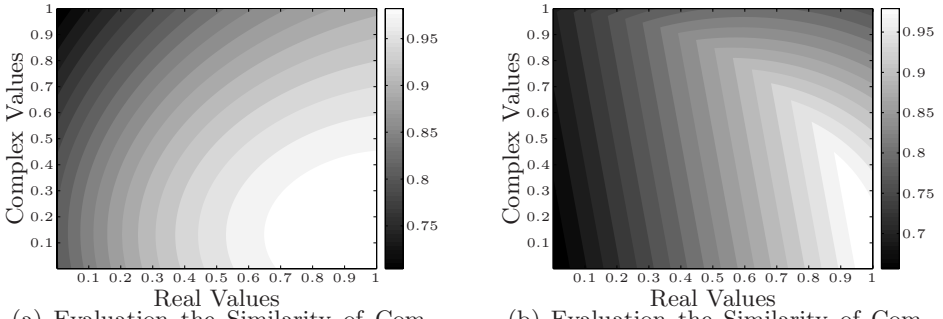
```
Evaluate sig_self(t)
Send sig_self(t) to random peer
Receive sig_peer(t − x) from peers
for All sig_peer(t − x) received do
  if rand > C3_self then
    R = match(sig_peer(t − x), sig_self(t))
    if rand < R then
      Signatures Match Inactivate Process
      if C3_self + C3_inc < C3_max then
        C3_self = C3_self(t) + C3_inc
      end if
    else
      if C3_self − C3_dec > 0 then
        C3_self = C3_self(t) − C3_dec
      end if
      Store sig_peer(t − x) and sig_self(t)
    end if
  else
    Add sig_peer(t − x) to outbound queue
  end if
end for
for All sig_peer(t − x) in outbound queue do
  if (t − x)) > T_MaxSig then
    Delete signature
  else
    Send sig_peer(t − x) to random peer
  end if
end for
```

The modified version is shown in (3). This removes the similarity in value for similar shapes of different magnitude and produces a slightly improvement in the ability to distinguish different vectors shown in figure 2(b).

$$R_2 = \frac{|<a,b>|}{max(\|a\|_2^2, \|b\|_2^2)} \tag{3}$$

Although the corrected matching functions perform well they remain relatively insensitive to the movement of one or two poles. Therefore it would require a large movement from the complete vector to bring about a low matching score. In order to produce a matching function more sensitive to changes in pole locations then a factor can be introduced that provides a non-linearity. This modification is shown in (4).

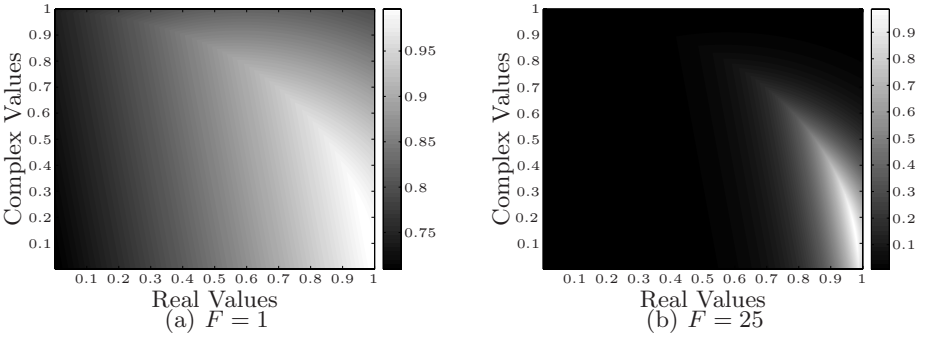
$$R_3 = e^{FR_2 - F} \tag{4}$$



(a) Evaluation the Similarity of Complex Vectors using (2)

(b) Evaluation the Similarity of Complex Vectors using (3)

Fig. 2. Effectiveness of Basic Cauchy Equations shown in (2) and (3)



(a) $F = 1$

(b) $F = 25$

Fig. 3. Affect of Varying F in (4)

In order to produce a higher variability in the result of the match an exponential function is introduced to give a highly localised comparison. In order that result of the function is bound between unity and zero an additional factor F is used to limit the swing of $FR_2 - F$ from zero to $-F$. In addition it allows the matching function to have a variable component that allows the shape of the matching region to be adjusted as suited to the problem area. Figures 3(a) and 3(b) shows the evaluation of (4) over the unit circle in the complex plane. As can be seen (4) is sensitive to changes in the evaluation of R_2, but can be adjusted by varying F to give a sharper region of suppression shown in Figures 3(a) and 3(b).

5 Conclusion

In this paper we have explored the possibility for an algorithm based on the Alternative Pathway of Innate Immune system. The Alternative Pathway is triggered by all cell surfaces. However, the surface of safe body cells strongly inhibit the process by attracting Factor H. The algorithm is based around a network

of homogeneous agents sharing information on how they are reacting to their environment. This information is akin to the picture presented by a cell surface in the Alternative Pathway. Further work is planned to exploit the algorithm in the fallible swarm problem to remove rogue elements.

References

1. Schaust, S., Szczerbicka, H.: Artificial immune systems in the context of misbehavior detection. Cybernetics and Systems 39(2), 136–154 (2008)
2. Wittig, T., Jennings, N.R., Mamdani, E.H.: ARCHON - A framework for intelligent cooperation. IEE-BCS Journal of Intelligent Systems Engineering - Special Issue on Real-time Intelligent Systems in ESPRIT 3(3), 168–179 (1994)
3. Festinger, L.: A theory of social comparison processes. Human Relations 7, 117–140 (1954)
4. Kaminka, G.A., Tambe, M.: Social comparison for failure detection and recovery. In: Agent Theories, Architectures, and Languages, pp. 127–141 (1997)
5. Kaminka, G.A., Pynadath, D.V., Tambe, M.: The role of agent-modeling in agent robustness. In: Proceedings of the conference on AI meets the real-world (1998)
6. Kaminka, G.A., Tambe, M.: What is wrong with us? improving robustness though social diagnosis. In: Proceedings of the Fifteenth National Conference on Artificial Conference on Artificial Intelligence AAAI 1998 Tenth Conference on Innovative Applications of Artificial Intelligence, pp. 97–104 (1998)
7. Thurman, J.M., Holers, V.M.: The central role of the alternative complement pathway in human disease. Journal of Immunology 176, 1305–1310 (2006)
8. Köhl, J.: The role of complement in danger sensing and transmission. Immunologic Research 34(2), 157–176 (2006)
9. Köhl, J.: Self, non-self and danger: A complementary view. Advances in Experimental Medicine and Biology 586, 71–94 (2006)
10. Thielens, N.M., Tacnet-Delorme, P., Arlaud, G.J.: Interaction of c1q and mannan-binding lectin with viruses. Immunobiology 205, 563–574 (2002)
11. Kerr, M.A.: The human complement system: Assembly of the classical pathway c3 convertase. The Biochemical Journal 189, 173–181 (1980)
12. Janeway, C., Travers, P., Walport, M., Schlomchik, M.: Immunobiology - The Immune System in Health and Disease, 6th edn. Garland Science Publishing (2005)
13. Degn, S.E., Thiel, S., Jensenius, J.C.: New perspectives on mannan-binding lectin-mediated complement activation. Immunobiology 212, 301–311 (2007)
14. Endo, Y., Takahashi, M., Fujita, T.: Lectin complement system and pattern recognition. Immunobiology 211, 283–293 (2006)
15. Wallis, R.: Structural and functional aspects of complement activation by mannose-binding protein. Immunobiology 205, 433–445 (2002)
16. Hajela, K., Kojima, M., Ambrus, G., Wong, K.H.N., Moffatt, B.E., Ferluga, J., Hajela, S., Gál, P., Sim, R.B.: The biological functions of MB-associated serine proteases (MASPs). Immunobiology 205, 467–475 (2002)
17. Wallis, R.: Interactions between mannose-binding lectin and MASPs during complement activation by the lectin pathway. Immunobiology 212, 289–299 (2007)
18. Walport, M.J.: Complement - first of two parts. New England Journal of Medicine 344(14), 1058–1066 (2001)
19. Pangburn, M.K., Müller-Eberhard, H.J.: The alternative pathway of complement. Springer Seminars in Immunopathology 7, 163–192 (1984)
20. Nicol, P.A.E., Lachmann, P.J.: The alternative pathway of complement activation. The role of C3 and its inactivator (KAF). Immunology 24, 259–265 (1973)

21. Müller-Eberhard, H.J.: Molecular organization and function of the complement system. Annual Reviews of Biochemistry 57, 321–347 (1988)
22. Whaley, K., Ruddy, S.: Modulation of the alternative complement pathway by β1H globulin. Journal of Experimental Medicine 144, 1147–1163 (1976)
23. Weiler, J.H., Daha, M.R., Austen, K.F., Fearon, D.T.: Control of the amplification convertase of complement by the plasma protein β1H. Proceedings of the National Academy of Science USA 73(9), 3268–3272 (1976)
24. Pangburn, M.K., Müller-Eberhard, H.J.: Complement c3 convertase: Cell surface restriction of β1H control and generation of restriction on neuraminidase-treated cells. Proceedings of the National Academy of Science USA 75(5), 2416–2420 (1978)
25. Kazatchkine, M.D., Fearon, D.T., Austen, K.F.: Human alternative complement pathway: Membrane associated sialic acid regulates the competition between B and β1H for cell-bound C3b. Journal of Immunology 122(1), 75–81 (1979)
26. Hourcade, D.E.: The role of properdin in the assembly of the alternative pathway c3 convertases of complement. The Journal of Biological Chemistry 281, 2128–2132 (2006)
27. Zipfel, P.F., Mihlan, M., Skerka, C.: The alternative pathway of complement: A pattern recognition system. Advances in Experimental Medicine and Biology 598, 80–92 (2007)
28. Nilsson, U.R., Müller-Eberhard, H.J.M.: Isolation of β_{1F}-globulin from human serum and its characterization as the fifth component of complement. Journal of Experimental Medicine 122, 277–298 (1965)
29. de Córdoba, S.R., Esparza-Gordillo, J., de Jorge, E.G., Lopez-Trascasa, M., Sánchez-Corral, P.: The human complement factor H: Functional roles, genetic variations and disease associations. Molecular Immunobiology 41, 355–367 (2004)
30. Alexander, J.J., Quigg, R.J.: The simple design of complement factor H: Looks can be deceiving. Molecular Immunology 44, 123–132 (2007)
31. Fearon, D.T.: Regulation by membrane sialic acid of β1H-dependent decay-dissociation of amplification C3 converase of the alternate complement pathway. Proceedings of the National Academy of Science USA 75, 1971–1975 (1978)
32. Meri, S., Pangburn, M.K.: Discrimination between activators and nonactivators of the alternative pathway of complement: Regulation via a sialic acid polyanion binding site on factor h. In: Proceedings of the National Academy of Sciences of the United States of America, vol. 87, pp. 3982–3986 (1990)
33. Pangburn, M.K., Schreiber, R.D., Trombold, J.S., Müller-Eberhard, H.J.: Paroxysmal nocturnal hemoglobinuria: Deficiency in factor h-like functions of the abnormal erythrocytes. Journal of Experimental Medicine 157, 1971–1980 (1983)
34. Favoreel, H.W., de Walle, G.R.V., Nauwynck, H.J., Pensaert, M.B.: Virus complement evasion strategies. Journal of General Virology 84, 1–15 (2003)
35. Pangburn, M.K.: Host recognition and target differentiation by factor H, a regulator of the alternative pathway of complement. Immunopharmacology 49, 149–157 (2000)
36. Pangburn, M.K., Pangburn, K.L.W., Koistinen, V., Meri, S., Sharma, A.K.: Molecular mechanisms of target recognition in an innate immune system: Interactions among factor H, C3b and target in the alternative pathway of human complement. The Journal of Immunology 164, 4742–4751 (2000)
37. Bellifemine, F., Poggi, A., Rimassa, G.: JADE - a FIPA-compliant agent framework. In: Proceedings of Practical Application of Intelligent Agents and MultiAgents (PAAM 1999), pp. 97–108 (1999)
38. Kreyszig, E.: Advanced Engineering Mathematics. John Wiley & Sons Inc., Chichester (1999)

Adaptable Lymphocytes for Artificial Immune Systems

Paul S. Andrews[1] and Jon Timmis[1,2]

[1] Department of Computer Science, University of York, UK
[2] Department of Electronics, University of York, UK
{psa,jtimmis}@cs.york.ac.uk

Abstract. The adaptable lymphocyte hypothesis is identified as a possible source of inspiration for artificial immune systems. Based on a number of qualitative investigations we identify some properties of a theoretical system (the tunable activation threshold model and excitability) that could be applicable in an engineering domain. An example is shown of how we could exploit these properties.

1 Introduction

The development of the majority of biologically inspired algorithms, such as artificial immune systems (AIS), has been criticised [1] for lacking a rigorous methodology to ensure that the algorithms are actually based on the biological properties from which they have been inspired. To investigate this criticism, we are following a process suggested by [1] to develop an AIS based on novel immune ideas that tries to capture the essence of the immunology. In particular, this paper focuses on one way in which we can take a theoretical immune property, the adaptable lymphocyte hypothesis [2,3], and investigate how it could be used in an engineering application.

In section 2 we describe some of our relevant previous work, then in section 3 the adaptable lymphocyte hypothesis is outlined. Sections 4 and 5 provide an investigation of the suitability for engineering of the adaptable lymphocyte hypothesis. Section 6 highlights a way in which we might use the adaptable lymphocyte hypothesis ideas for population dynamics in an AIS, and in section 7 we draw conclusions on our work.

2 Previous Work

In [4], we explored how AIS have been developed in recent years focusing on their immunological inspirations. This led us to suggest that actively seeking out new immune theories for providing AIS inspiration could be of benefit. One such theory was identified as Cohen's [5] cognitive immune system, which was shown to incorporate many appealing properties that could inspire an engineering system, such as degeneracy and patterns of response. We also identified in [4] the conceptual framework approach of Stepney et al. [1] as a methodology for exploiting a new

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 376–386, 2008.

immune theory from which you wish to take inspiration for building an AIS. This framework aims to facilitate the development of bio-inspired algorithms in a more principled way than has been previously observed. It suggests AIS are designed through a series of observational and modelling stages in order to identify the key characteristics of the immunological process on which the AIS will be based. The first stage probes the biology using observations and experiments to produce a partial view of the biological system under investigation. From this view, abstract models of the biology are built, which are open to investigation and validation techniques not available to the actual biological system. The insight gained from these models should then lead to the construction of the bio-inspired algorithms and frameworks. It is important to note that the process is iterative and allows movement between stages depending on their outcomes. It is suggested that algorithms developed in this way will be more biologically plausible and avoid being a weak analogy of the process on which they are based having being developed directly from (often naive) biological observations.

We have chose to follow the conceptual framework approach in an attempt to develop a novel AIS, and to investigate the claim that algorithms developed in this way are more biologically plausible. We began our investigations in [6] with a model and simulation of degeneracy in a lymph node inspired by the ideas of Cohen's [5] holistic view of the immune system. Our partial view of the biology was taken directly from the immunological research literature, from which a model was extracted. This model was based on the process of T_H cell activation in the paracortex of a lymph node, in which the T_H cell receptors are assumed to be degenerate. Our results highlighted the ability of randomly generated detectors to collectively produce distinct patterns of response to antigenic stimuli. However, one of the main outcomes of this work was that additional immunology was needed to inspire a useful engineering application. Notably, we were interested in finding a mechanism of adaptivity in T cells that could result in interesting population dynamics that we could exploit in an AIS.

3 The Adaptable Lymphocyte

In response to a number of observations that contradict the classical view of immunological tolerance, Grossman [2,3] presents the adaptable lymphocyte hypothesis: the "responsiveness of individual lymphocytes to antigen and other signals can be tuned and updated" [3]. From this hypothesis, Grossman [2] derives the tunable activation threshold (TAT) model that assumes lymphocyte activation thresholds are tuned internally by the cell based on the history of its environmental stimulation. The key definitions from the TAT model are:

Excitation. a quantitatively expressed change in the metabolic state of a cell induced as a direct result of an external stimulus.

Excitation Level. a positive scalar measure of the excitation.

Excitation Index. a time dependent, weighted average of the past excitation levels of the cell.

Perturbation. the difference between the current excitation level and the excitation index upon an excitation event.
Activation Threshold. the excitation index plus a fixed critical value.

Grossman [2] suggests that the excitation index at time t, $I(t)$, could be related to the excitation, $E(t)$ via equation (1):

$$\frac{dI(t)}{dt} = \alpha E(t)[E(t) - I(t)] \tag{1}$$

where α is a positive constant.

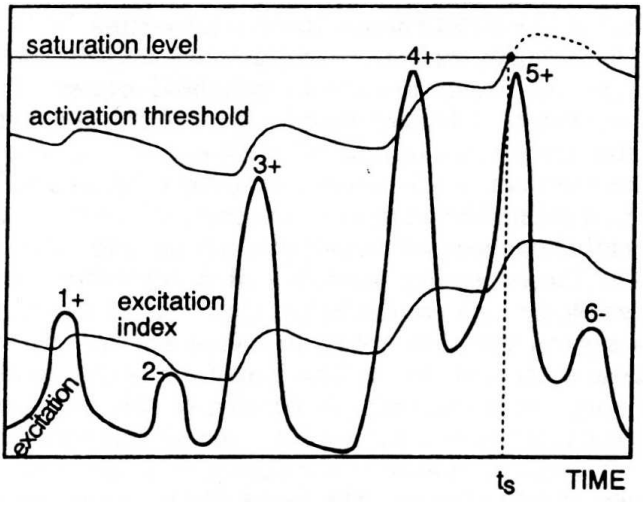


Fig. 1. TAT model behaviour reproduced from [2]

Examples of the dynamics of the TAT model are shown in Figure 1, which has been reproduced from [2]. This shows six distinct perturbations to the excitation of a cell over a period of time with the corresponding evolution of the excitation index and activation threshold are traced. Each perturbation is labelled along with a sign (+ or −) highlighting whether the excitation has breached the excitation index. For activation to occur, the excitation level must exceed the activation threshold. Of the six events shown, only the fourth would lead to activation. At time t_s, activation can no longer occur as the activation threshold has exceeded the saturation level, which is defined as the cell becoming temporarily anergic.

In addition to the TAT model, Grossman [3] incorporates tunable excitability into the adaptable lymphocyte hypothesis, defining it as "a measure of the cell's capacity to communicate with other relevant cells". It is described as being directly promoted by excitation events and an enhanced excitability facilitates the ability of the cell to proliferate and differentiate. It is further noted that the kinetics of the excitation index and excitability determines whether a lymphocyte will be activated.

The adaptable lymphocyte hypothesis has a number of appealing properties from an engineering point of view. This has also been identified by Guzella et al. [7] who suggest using tunable T cell thresholds as part of an AIS aimed at temporal anomaly detection. They highlight an initial architecture for a T cell inspired anomaly detection system attempting to incorporate previously non-utilised aspects of the natural immune system. They focus on the signalling machinery of T cells and the TAT model, although the TAT model is not yet integrated into their architecture. Guzella et al. [7] also highlight the work of a number of TAT models from theoretical immunology [8,9,10]. Unlike these works we are not concerned with whether the TAT equation (1), or any other more complicated equation, is biologically accurate, but how we can exploit Grossman's [2,3] ideas for AIS.

4 Investigating TAT Behaviours

The purpose of the investigations that follow is to first assess whether the TAT equation (1) gives us the behaviour stated by Grossman [2,3] and summarised in section 3, and then to investigate how we might use this in an engineering context. Our goal is to translate the qualitative behaviour seen in the TAT model into an engineering domain. The approach we have taken is to build a simple computational model that we have used for the examples presented in the next two sections. The model comprises two main components, a population of *detectors* with tunable excitation indexes and a population of *antigens* used to stimulate the detectors and provide the excitation. The model is iterative and at each time step, t, each detector is exposed to an antigenic stimulus and the following equation (2) (equivalent to equation (1)) is used to update its excitation index:

$$I(t+1) = I(t) + \alpha E(t)[E(t) - I(t)] \tag{2}$$

where the symbols are the same as those described for equation (1). The way in which the detectors and antigens interact differs for each example that follows.

4.1 The TAT Equation

Here we look at each component of the update equation (2) to assess its effect on the evolution of the excitation index. At each time step, the excitation index is increased by the multiplication of three terms, a positive constant α, the excitation $E(t)$, and the size of the perturbation to the system $[E(t) - I(t)]$. This perturbation is the only term that can be negative, thus a positive perturbation will lead to an increase in the excitation index and a negative perturbation to a decrease. The larger the perturbation, the larger this term will be. The α parameter determines how quickly the excitation index tunes, the smaller it is, the smaller the increment to the excitation index and the slower it tunes to the value of the excitation. This parameter, therefore, controls the "memory effect" of the excitation index: the lower the α value the longer term memory of past excitations.

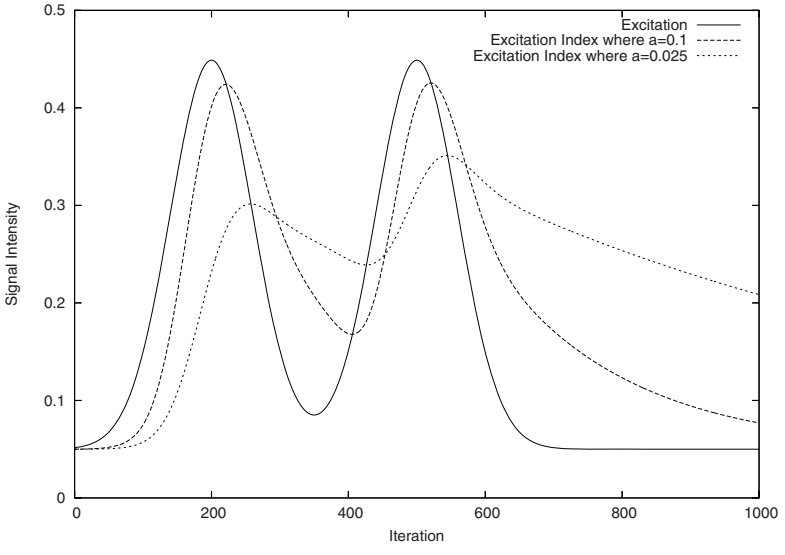


Fig. 2. Effect of parameter α on the excitation index with a varying excitation

Figure 2 demonstrates how the excitation index of two tunable detectors with different α parameters (0.1 and 0.025 respectively) changes given a varying excitation. The excitation distribution was generated by two gaussian distributions centred on iterations 200 and 500. These values were then applied to a detector as the antigenic stimulus over time. This figure shows the excitation index is behaving as described in section 3, being tuned by the excitation events that are occurring. We can also see the effect of α with at lower value producing an excitation index curve that adapts (tunes) at a slower rate.

Figure 2 also highlights the effect of the third term that influences the excitation index increment, the $E(t)$. This provides the behaviour whereby the excitation index tunes more quickly at higher excitations. It is this term that makes the TAT equation (1) differ from a simple sliding average of previous excitations. As the excitation value falls, so does the tuning of the excitation until at $E(t) = 0$ where no tuning occurs. In a biological setting, this lack of stimulus may be unlikely to occur, however in an engineering context we need to be aware of this behaviour as it could be an unwanted property if a constant background level of excitation can't be guaranteed.

As a final example, we generate an excitation distribution to visually match the original figure of Grossman [2] (reproduced here in Fig. 1) to see if we could re-create the behaviours of the excitation index and activation threshold. The result is shown in Fig. 3, which provides a good visual match to Fig. 1. We conclude from these observations that the TAT equation (1) is suitable to produce qualitatively similar behaviours as those described by Grossman [2,3] and summarised in section 3.

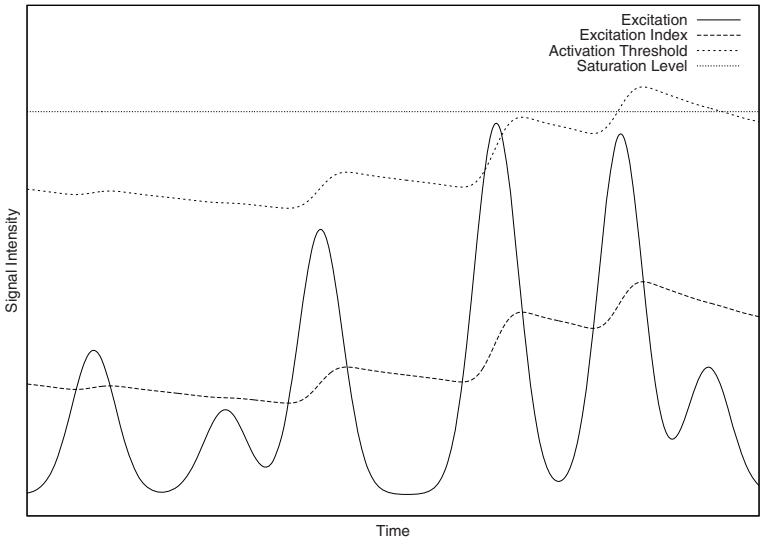


Fig. 3. Reproduction of TAT behaviours graph using equation (1) and $\alpha = 0.0075$

4.2 Simple Engineering Example

Our next step assessing the suitability of the TAT model for engineering, was to investigate the behaviour with a detector and antigen representation similar to those used in an AIS. Typically this involves providing each with a string of symbols (normally binary or real numbers) that represent the molecular shape of their binding regions. We therefore gave our detectors and antigens a vector of real numbers, which is used to calculate their affinities based on their Euclidean distance. This affinity is then multiplied by a concentration for the antigen to transform it into an avidity measure, representing the excitation of a detector at that point in time. For this example, we subjected two detectors with different shapes and the same α parameter to the same antigen over a period of 1000 iterations. For the first 300 iterations, the antigen concentration was set at 20 to provide a background level of excitation. For the next 200 iterations, the antigen concentration was 200 mimicking an excitation event. After this, the concentration falls back down to 20. The results for an antigen with the shape $[0.7, 0.7, 0.7, 0.7, 0.7]$, detector 1 with $[0.25, 0.25, 0.25, 0.25, 0.25]$ and detector 2 with $[0.25, 0.25, 1.0, 0.5, 0.5]$ are presented in Fig. 4. This shows a qualitatively similar excitation index tuning behaviour to the more abstract examples above in section 4.1. This re-enforces our view that the TAT equation (1) can be used effectively in an engineering setting.

5 Excitability

As noted in section 3, Grossman [3] describes the property of excitability as part of the adaptable lymphocyte hypothesis. Unlike with the TAT model, no

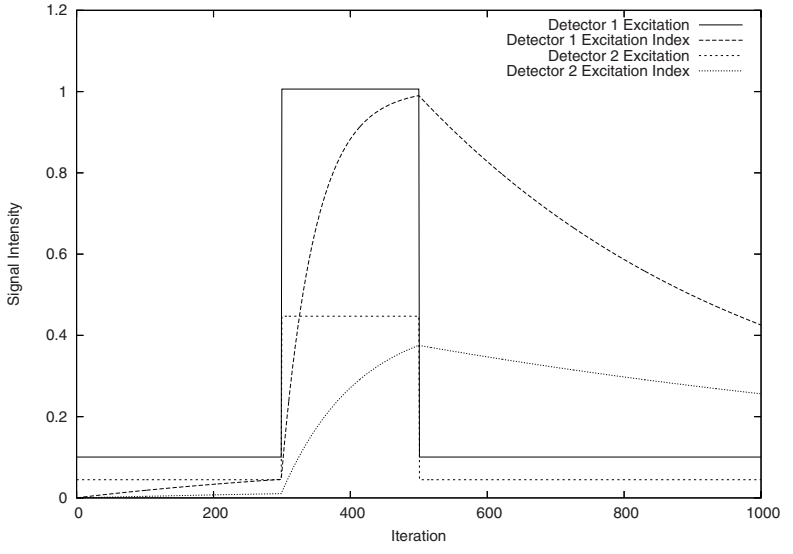


Fig. 4. Excitation level and index of two different randomly generated detectors exposed to an antigen

equation is given to describe the behaviour of excitability, only that it is promoted by excitation events and facilitates the ability of the cell to proliferate and differentiate. Along with the excitation index, it determines whether a lymphocyte will be activated or not. Thus, if we are to use the ideas of the adaptable lymphocyte hypothesis in an engineering system, then an equation to describe excitability could be advantageous.

Based on the descriptions above and the dynamics of the TAT equation (1), we propose that the following equation (3) can be used to describe the excitability, $X(t)$:

$$X(t+1) = \begin{cases} X(t) + \sigma[E(t) - I(t)][\mu - X(t)] & \text{if } E(t) - I(t) > 0 \\ \delta X(t) & \text{otherwise} \end{cases} \quad (3)$$

where t is time, $E(t)$ is the current excitation, $I(t)$ is the excitation index and σ, μ and δ are all positive constants. This equation (3) is split into two cases: the first applies when there is a positive perturbation to the system, otherwise the second part is applied. This second case simply implements a decay of the excitability over time if there are no positive excitation events, with δ acting as a parameter that determines how fast this decay happens. The first case of the equation is influenced by equation (1), and attempts to take into account the idea that a positive perturbation will promote excitability, hence the inclusion of the $[E(t) - I(t)]$ term. σ acts as a parameter that determines how quickly the excitability tunes, providing a similar behaviour to the α parameter of equation (1). The last term, $[\mu - X(t)]$, provides a scaling to the excitability, where μ defines a maximum excitability for the detector.

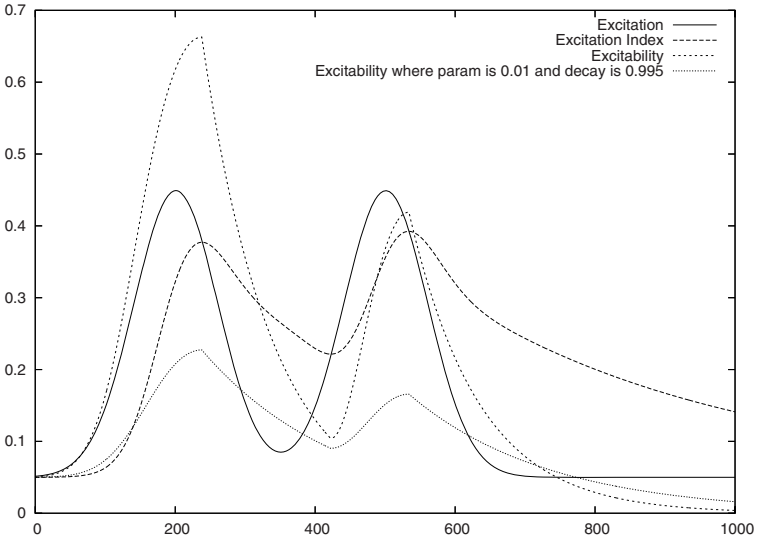


Fig. 5. Typical excitability graph

Using the same excitation distribution in Fig. 2, Fig. 5 plots the excitability for two detectors with different σ and δ values, but the same value for μ. Both detectors are also given the same values for the excitation index, allowing us to compare the effects of σ and δ on excitability. As with α, we see that at lower values of σ, the excitability tunes more slowly. Conversely, the larger the value of δ the slower the excitability decays. We also see the behaviour difference between the excitation index and the excitability. Both have similar qualitative behaviours in the presence of a positive excitation event (controlled by their respective tuning parameters α and σ). Their behaviour differs however in the presence of a decreasing excitation event, and in the absence of excitation. Instead, the excitability decays at a controllable rate determined by δ and will continue to tune downwards in the presence of no excitation.

6 Population Patterns

Having investigated the behaviours of the TAT model and a possible equation for excitability in terms of their suitability for an AIS, we can start to examine how to integrate tunable detectors into ideas from our previous work. In [4,6] we have identified and investigated the idea of using patterns of a population of degenerate detectors to provide immune specificity. In this section we show one way in which we can augment this idea by adding adaptability to the detectors.

The excitation index and the excitability provide similar, but different dynamics, each in some way providing a memory of previous excitation events. We propose that these can be used in combination to control the proliferation of detectors undergoing a continued antigenic stimulus. Once the excitation level of

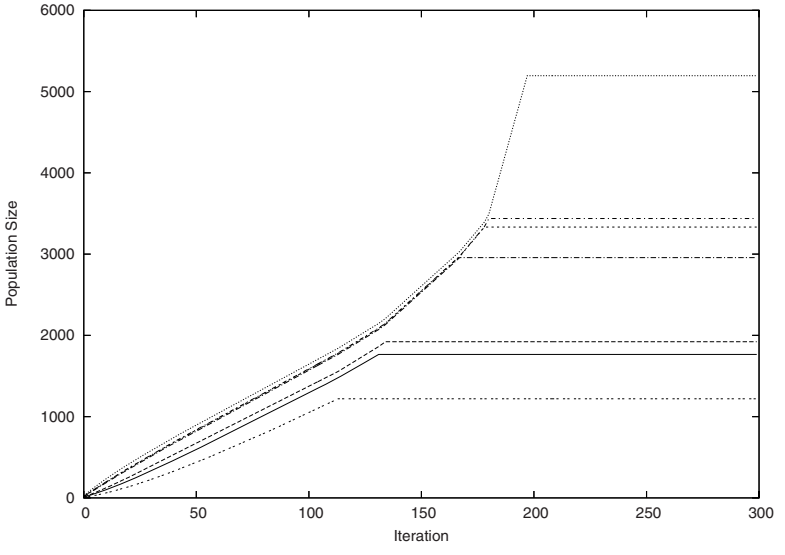


Fig. 6. Population response of 7 detectors to a continued antigen presence

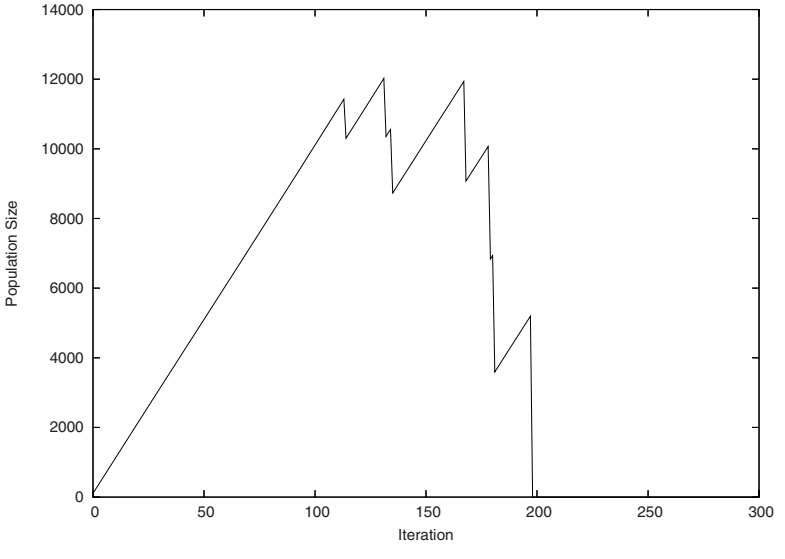


Fig. 7. Total population pattern of active cells

a detector (determined by its avidity for the antigenic stimulus) rises above the tuned activation threshold, the excitability is used to determine the probability that it will proliferate. The higher the excitability, the larger this probability would be. As the excitation index continues to tune, a point will occur when the excitation level is no longer above the activation threshold and the clonal detector population will stop expanding in size. Thus, the size of this population

is determined by a function of the avidity of the detector for the antigen present and the interplay between the excitation index and excitability. Given a randomly generated population of these detectors we will get a clonal expansion of the detectors occurring at different rates, producing a pattern of population response that is typical of the antigen that induced it. As an example, Fig. 6 shows the populations of 7 randomly generated detectors expanding to the same antigenic stimulus. For each detector the point at which it stops expanding is different, and the sizes of the clonal population are also different. If we look at total population of all detectors and plot the numbers of active cells (i.e. those that are still expanding) we get the graph shown in Fig. 7. This provides us with a single pattern of response for the entire population. It is this pattern that could be incorporated into an AIS.

7 Conclusions

From the investigations presented in sections 4, 5 and 6 there are a number of implications we can draw for designing AIS that would take advantage of tunable detectors based on the adaptable lymphocyte hypothesis. Based on a number of qualitative investigations, we have highlighted a way in which we can translate properties of a theoretical system into behaviours that might be useful for engineering. Additionally the process of investigating the TAT model and excitability give us an understanding of the dynamics of the tuning equations and the effects of their parameters. It is likely that an AIS based on these ideas will have similar behaviours and parameters, thus our investigations would allow us to better reason about the AIS.

In summary, we have followed on from previous work [4,6] aimed at following the conceptual framework approach [1] to develop a novel AIS, and identified the adaptable lymphocyte hypothesis of Grossman [2,3] as a possible source of AIS inspiration. We have presented a qualitative investigation into the TAT model and deemed it suitable for use in an engineering context. We have also presented an equation (3) for describing the property of excitability. Using a combination of the TAT model and excitability we have given an example of how the adaptable lymphocyte ideas can be used to provide a population response from a set of detectors. This process of investigation has given us insight into how we can translate an immune theory into a property useful in an engineering. Further (as yet unpublished) work has uses these ideas in an AIS for pattern classification using patterns of a population of degenerate and adaptable lymphocytes.

References

1. Stepney, S., Smith, R., Timmis, J., Tyrrell, A., Neal, M., Hone, A.: Conceptual frameworks for artificial immune systems. International Journal of Unconventional Computing 1(3) (2005)
2. Grossman, Z., Paul, W.E.: Adaptive cellular interactions in the immune system: The tunable activation threshold and the significance of subthreshold responses. Proceedings of the National Academy of Sciences (PNAS) 89, 10365–10369 (1992)

3. Grossman, Z.: Cellular tolerance as a dynamic state of the adaptable lymphocyte. Immunological Reviews 133, 45–73 (1993)
4. Andrews, P.S., Timmis, J.: Inspiration for the next generation of artificial immune systems. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 126–138. Springer, Heidelberg (2005)
5. Cohen, I.R.: Tending Adam's Garden: Evolving the Cognitive Immune Self. Elsevier Academic Press, Amsterdam (2000)
6. Andrews, P.S., Timmis, J.: A computational model of degeneracy in a lymph node. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 164–177. Springer, Heidelberg (2006)
7. Guzella, T.S., Mota-Santos, T.A., Caminhas, W.M.: Towards a novel immune inspired approach to temporal anomaly detection. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 119–130. Springer, Heidelberg (2007)
8. Carneiro, J., Paixão, T., Milutinovic, D., Sousa, J., Leon, K., Gardner, R., Faro, J.: Immunological self-tolerance: Lessons from mathematical modeling. Journal of Computational and Applied Mathematics 184, 77–100 (2005)
9. van den Berg, H.A., Rand, D.A.: Dynamics of T cell activation threshold tuning. Journal of Theoretical Biology 228, 397–416 (2004)
10. Scherer, A., Noest, A., de Boer, R.J.: Activation-threshold tuning in an affinity model for the T-cell repertoire. Proceedings of The Royal Society B 271(1539), 609–616 (2004)

On the Relevance of Cellular Signaling Pathways for Immune-Inspired Algorithms

T.S. Guzella[1,2] and T.A. Mota-Santos[2]

[1] Dept. of Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte (MG) 31270-010, Brazil
tguzella@cpdee.ufmg.br
[2] Dept. of Biochemistry and Immunology, Federal University of Minas Gerais, Belo Horizonte (MG) 31270-010, Brazil
tomaz@icb.ufmg.br

Abstract. In this conceptual paper, we discuss the relevance of cellular signaling pathways for immune-inspired algorithms. With complex dynamics, the mapping of environment stimuli to cellular responses is highlighted as a decision making capability. When considering applications which could benefit from these dynamics, the possibility of incorporating these pathways can be an interesting way to combine more biologically-plausible algorithms and improved performance. The structure of the NF-κB (Nuclear Factor κB) and MAP (Mitogen-activated protein) kinases pathways, and the pathways involved in signaling by Toll-like receptors, are presented. As an example, we then consider how these pathways could be incorporated in the Dendritic Cell Algorithm.

Keywords: Artificial Immune Systems, Signaling pathways, NF-κB, MAP kinases, Toll-like receptor signaling.

1 Introduction

Nature has always been an interesting source of inspiration for engineers and computer scientists. In recent years, it has lead to the proposal of important computational tools, such as Artificial Neural Networks and Genetic Algorithms. Based on the powerful cognitive capabilities of the human immune system, a more recent development are Artificial Immune Systems [1] (AISs). AISs have been used in various application areas, inspired by several processes taking place in the immune system.

The immune system is often cited as possessing several interesting features from a computational perspective, such as pattern recognition, memory, homeostatic stability, among others. However, in attempting to capture such features in an algorithm, it should be kept in mind that they are "implemented", at the cellular level, by signaling pathways. This is further emphasized by the increasing application of mathematical formulations to signaling dynamics in the immune system [2], with the objective of unveiling their roles. Due to the fact that many of these features of interest can be described as emerging from such pathways,

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 387–398, 2008.

along with inter-cellular interactions taking place in the immune system, understanding how these pathways are organized can have important consequences. Thus, considering some of these features in a specific application might require the understanding, at some level of detail (even if very simplistic), of how these pathways shape the response of cells. From a conceptual point of view, this requires looking at complexity at the cellular level, in addition to that at the population level. This constitutes the main point of this paper: to discuss the relevance of considering these pathways in immune-inspired algorithms, and to suggest how this could be done. In line with our argument of the importance of understanding the structure of these signaling pathways, we presented a reasonably detailed description of some of them.

In particular, in the area of neural networks, the importance of the dynamics of neurons, due mainly to processes involving ion channels, has been receiving increasing attention. In contrast to the Multi-layer perceptron (MLP), which assumes that the information transmitted between two neurons is coded in the average spiking rate, so that the neuron's output is a smooth function of the input, several recent models include the dynamics involved in the generation of spikes. In addition to being more biologically realistic, such models have a wide applicability in problems where dynamical aspects are important, such as sound analysis [3] and robotics [4].

When it comes to more biologically-plausible models, a similar change is occurring in AISs, with recent works advocating algorithms more realistic from a biological point of view [5]. However, in doing so, the characteristics of the algorithm should be tailored to the target application [6]. This process involves several steps, such as the understanding of the biological processes of interest, the construction of models for analysis of these processes and, then, the formulation of an algorithm [5], based on characteristics of the target application, thereby reinforcing the interdisciplinary characteristic of AISs [7].

This paper is organized in the following way: section 2 discusses the dynamics of signaling pathways, and the emergence of decision making capabilities. In sequence, section 3 presents two relatively well-known signaling pathways, the NF-κB (Nuclear Factor κB) and MAPK (Mitogen-activated protein kinases), followed by section 4, where the pathways involved in signaling by the Toll-like receptors are considered. Section 5 then considers, as an example, how signaling dynamics could be incorporated in the Dendritic Cell Algorithm [8,9], followed by the final conclusions and future research directions in section 6.

2 Dynamics of Signaling Pathways

Signaling pathways are ubiquitous in several cell types, not limited to cells in the immune system. They allow a cell to adapt in response to certain environmental stimuli. A general description of the steps involved in the activation of such pathways is shown in figure 1. The initial event is the binding of a ligand, such as cytokines, hormones, an antigen or a peptide-MHC complex, to membrane receptors. This activates downstream events taking place in the cytoplasm,

which, usually (but not only) through possibly multiple phosphorylation or dephosphorylation (addition/removal of a phosphate group, respectively) steps of one or more proteins, mediated by kinases and phosphatases, respectively, leads to the generation of one or more multi-protein complexes. As indicated in figure 1, some of the steps involved in a certain pathway may be shared with another pathway, for example when an enzyme needed for the activation of one pathway also mediates activation (or even inhibition steps) in another pathway. At some point, the nuclear transport of certain species formed (or activated) during the activation of the pathway takes place, where they influence the expression of certain genes. These genes can, in turn, lead to the expression of proteins involved in one or more pathways or the secretion of soluble factors. In the former case, the resultant proteins might up- or down-regulate the activated pathway, through interaction with the receptors or by regulating some of the steps during the activation of the pathway. It is also possible that the affected genes induce the up- or down-regulation of the membrane receptors. In turn, the secreted factors can stimulate another or even the same type of receptor initially activated, resulting in the activation of other (or the same) pathway.

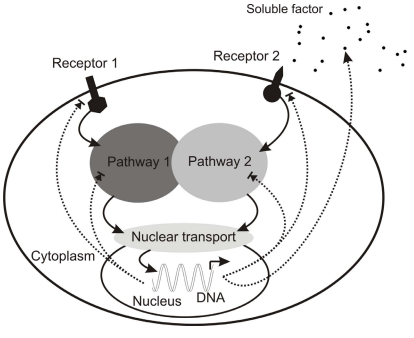


Fig. 1. Illustration of some of the general events involved in signaling pathways. The dotted arrows indicate events that occur in response to activation of a pathway (see text).

While the previous discussion might give an impression that the activation of such pathways is a linear, sequential, event, the realization that this is not the case is a growing theme in the literature. It is being increasingly acknowledged that the functioning of these pathways is extremely complex, due to the interconnection between components taking place in several pathways, challenging reasonably simple chain of events following activation. One characteristic of many of these pathways is combinatorial complexity, which results from the combinatorial number of complexes that can be formed as a result of ligand stimulation (e.g. [10]). In particular, this is one of the major obstacles for studying the signaling networks in a cell. Nevertheless, the large number of connections and the modularity [11] typical of such networks might allow them to perform complex input-output mappings [12], in terms of the cellular responses (proliferation, apoptosis, among others) to stimulation, highlighting the emerging information processing and decision-making capabilities of these networks [13].

Despite the difficulties involved in elucidating the molecular events taking place in signaling pathways, the growing availability of technologies allowing the study of such systems has allowed an increased understanding of such events. In this context, an important aspect studied is the structure of these signaling pathways, i.e. what are the components involved and how they are organized during the response. This information is extremely important, due to the fact that the structure of a pathway, represented by the interactions between components involved in such pathway, shape the cellular response to stimulation. An increasingly used tool to analyze this structure is mathematical modeling, which allows a characterization of the emerging features of these pathways. Among the approaches that have been used to model signaling pathways, the following can be cited (although this list is far from complete):

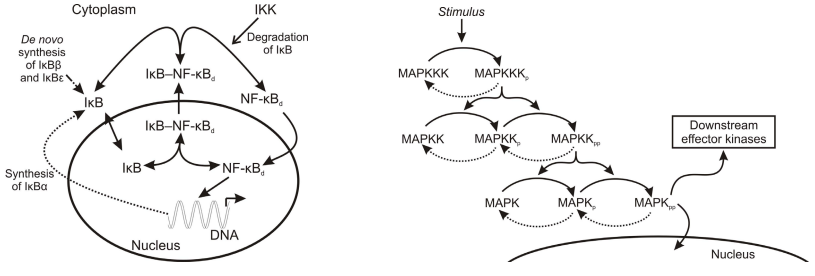
- Differential equations, either ordinary (where the spatial distribution of components is assumed to be homogeneous) or partial (which consider spatial aspects). Sometimes, stochastic effects are also considered.
- Rule-based models [13], reviewed in [14].
- Algebra-based models [15].
- P-Systems, also known as membrane computing [16].

3 The NF-κB and MAPK Signaling Pathways:

In this section, we briefly present two pathways involved, among other functions, in several aspects of the immune system: the NF-κB (Nuclear Factor κB) and the MAPK (Mitogen-activated protein kinases) pathways. This discussion serves two purposes: illustrating the complex dynamics of these two signaling pathways, while paving the road for the next section, as these two pathways are involved in signaling mediated by the family of Toll-like receptors (TLRs).

The NF-κB pathway, depicted in figure 2a, is believed to the "original signaling pathway" (see the final words in [17]), given the widespread expression in invertebrates of genes coding proteins involved in this pathway. In a stimulation-free scenario, the majority of NF-κB dimers (complexes formed by two molecules of NF-κB), indicated in figure 2a with the subscript d, are located in the cytoplasm, associated with one of three IκB proteins (IκB$\alpha/\beta/\varepsilon$), which precludes their nuclear translocation. The IκB proteins, on the other hand, constitutively translocate between the cytoplasm and the nucleus. Stimulation of the NF-κB pathway leads to the activation of IKK (IκB kinase), which triggers the phosphorylation and degradation of the IκB proteins, allowing the NF-κB dimers to reach the nucleus, where they regulate the activation of hundreds of genes (such as inflammatory genes), including the one coding IκBα. Once in the nucleus, the dimers require association with IκB proteins in order to be transported back to the cytoplasm. The dynamics of this pathway can be analyzed using, for example, the three-dimensional nonlinear model proposed by Krishna et al. [18]. These authors have argued that the emergence of periodic spikes in the nuclear concentration of NF-κB is associated with an increased sensitivity of the pathway, which could allow the differential regulation of certain genes.

The MAPK pathway is another evolutionarily conserved signaling pathway, featuring three pathways, mediated by JNK (JUN N-terminal kinase), p38 and ERK (extracellular-signal-regulated kinase), although a further distinction is usually made between some pathways involving ERK [19,20,21]. In the following discussion, we focus only on the first two pathways, as those have been implicated in signaling by Toll-like receptors. Each one of the MAP kinase pathways can be described in a general way as a three level cascade, as shown in figure 2b. The first level is formed by the MAPK kinase kinases (MAPKKK), which are activated by phosphorylation. The phosphorylated MAPKKKs mediate, in turn, the double phosphorylation of MAPKKs (MAPK kinases, also known as MKKs), which in a similar way, mediate the activation of MAPKs. In figure 2b, single- and double-phosphorylation are indicated by the *p* and *pp* subscripts, respectively. In these two levels, only the double-phosphorylated forms are capable of mediating the activation of downstream substrates. In addition, phosphatases mediate the de-phosphorylation of the activated species, shown in figure 2b as dotted arrows, with the expression of some of those phosphatases influenced by the activation of the MAPKs. Once activated, MAPKs mediate signaling in the cytoplasm or in the nucleus, leading to various responses, such as the production of pro-inflammatory cytokines, the induction of cellular differentiation or apoptosis. After nuclear translocation, they affect the activity of transcription factors. In the cytoplasm, they mediate the activation of downstream signaling pathways, through the activation of kinases such as MAPKAPK-2 (MAPK-activated protein kinases). The dynamics of a mathematical model [19] of the MAPK cascade indicate an ultra-sensitive (step-like) response and bistability. In addition, when considering a negative feedback loop induced by activated MAPK that de-activates the MAPKKK, sustained oscillations can arise.



a) NF-κB pathway b) MAPK pathway

Fig. 2. Structure of the NF-κB and MAPK pathways

4 The Toll-Like Receptor Signaling Pathway

The Toll-like receptors (TLRs) are one of the front-line mechanisms for the identification of pathogens by the innate immune system. These receptors, expressed by cells such as macrophages and dendritic cells (DCs), recognize specific molecular patterns, and are crucial in the early response to pathogenic microorganisms.

Recent results point out the existence of at least 12 mammalian TLRs [22], some of which are expressed on the surface of cells, while others are present in intracellular compartments. The former (TLRs 1, 2, 4, 5 and 6) recognize mainly bacterial products (such as the recognition of LPS, produced by Gram-negative bacteria, by TLR4) which are not made by the host, while the latter (TLRs 3, 7, 8 and 9) recognize nucleic-acid structures, which are not unique to pathogens, but are not accessible to TLRs under normal conditions [23,24]. In addition, each receptor is capable of recognizing several distinct ligands (e.g. in the case of TLR4, LPS, heat shock proteins, fibrinogen and others [24]).

TLRs occur as either homo or heterodimers (complexes formed by two either equal or different species, respectively), whose formation is ligand-independent. In the case of TLR2, it associates with either TLR1 or TLR6, while the remaining TLRs mainly occur as homodimers. Following ligand stimulation, the triggered TLRs recruit molecular adaptors, leading to the activation of downstream signaling cascades. Currently, five adaptors are known:

- MyD88 (Myeloid Differentiation Factor 88),
- TRIF (TIR-domain-containing adaptor protein inducing IFN-β),
- MAL (MyD88-adaptor-like protein), also known as TIRAP (TIR-containing adaptor protein),
- TRAM (TRIF-domain-containing adaptor molecule),
- SARM (Sterile α- and armadillo-motif-containing protein).

The currently held model of TLR signaling, shown in figure 3, features two main pathways, referred to as MyD88-dependent and -independent pathways, where the former is shared with IL-1 (a pro-inflammatory cytokine), and the latter is mediated by TRIF. The activation of these pathways depends on the stimulated receptors. TLR3 uses only the MyD88-independent pathway, while TLR4 uses both, with the remaining receptors activating the MyD88-dependent pathway.

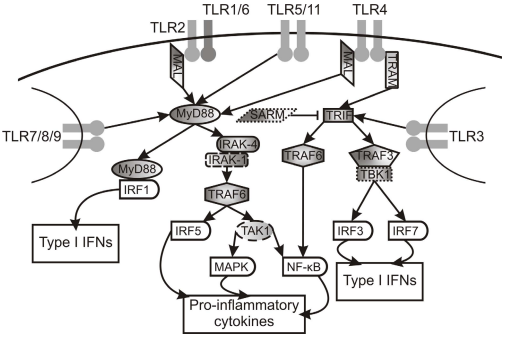


Fig. 3. Depiction of the TLR signaling pathway (see text for the description of the steps involved)

An important characteristic of the TLR pathway is that several steps involved in signal transduction are shared between several receptors. Therefore, a major question that arises is how response specificity is achieved, which can be described as the activation of specific genes as a result of the stimulation of certain receptors. Currently, it is believed that the specificity results from the extracellular and intracellular interactions between TLRs. The former is related to the dimerization of receptors, and the latter is due to the differential recruitment of adaptors following stimulation. In addition to the activation of the MyD88-dependent and TRIF-dependent pathways, TLRs 2 and 4 require the recruitment of MAL before MyD88 is recruited (see figure 3), while the recruitment of TRAM to TLR4 is necessary before the activation of the TRIF-dependent pathway can take place. Finally, SARM, whose expression is increased following TLR3/4 stimulation, inhibits downstream activation of the TRIF-dependent pathway.

Following MyD88 recruitment, IRAK-4 (IL-1R-associated kinase) is recruited and binds to MyD88. It then recruits and phosphorylates IRAK-1, associating with TRAF6 (TNF-receptor-associated factor 6). Following this, intermediate steps omitted in figure 3 involve the TRAF6-mediated activation of TAK1 (TGF-β-activated kinase), leading to the activation of the NF-κB and MAPK pathways, which induce the expression of genes encoding pro-inflammatory cytokines (such as TNF-α, IL-1β, IL-6 and IL-12). In case of the MAPK pathway, this is mediated by the activation of the JNK and p38 cascades by TAK1, which functions as a MAPKKK. In addition, following association with phosphorylated IRAK-1, TRAF-6 mediates the activation of IRF5 (Interferon regulatory factor 5), which also mediates the activation of pro-inflammatory genes. Another pathway mediated by MyD88 is the activation of IRF1, which requires the nuclear translocation of a MyD88-IRF1 complex, resulting in the temporary sequestration of MyD88, and the up-regulation of type I IFNs (IFN-α/β).

On the other hand, the TRIF-dependent pathway leads to the activation of TRAF6, and the recruitment of TRAF3, which results in the activation of TBK1 (TRAF-family-member-associated NF-κB-activator-binding kinase). TRAF6 activates the NF-κB pathway, up-regulating pro-inflammatory cytokines, while TBK1 activates IRF3 and IRF7, the latter only in plasmacytoid DCs, inducing the production of type I IFNs.

As a consequence of the intricate pathways involved in the signaling by TLRs, interesting emerging features in the TLR pathway are cooperation, synergism and antagonism, resulting from signaling from different receptors [22]. These features include the non-additive production of TNF following simultaneous stimulation of TLR2 and TLR4, the differential induction of genes resulting from the combined TLR3/TLR9 signaling [25], and the secretion of anti-inflammatory cytokines (such as IL-10) following TLR2 stimulation, which inhibit effects mediated by the subsequent stimulation with TLR3 or TLR4. Therefore, the combination of signals and their particular timing can have a profound influence on the cellular responses induced and the immune response.

5 Conceptualization of Signaling Pathways in Immune-Inspired Algorithms

As listed in the previous sections, the organization and dynamics of signaling pathways can have an important consequence in the functioning of the immune system. In this section, we look at the biological information presented and discuss how it may be incorporated into immune-inspired algorithms, taking the dendritic cell algorithm (DCA) [8,9] as an example. In doing so, we formulate a general agent-based representation of a cell, incorporating the dynamics of signaling pathways. As agent-based representations are widely used (e.g. [26,8]), this formulation should facilitate the incorporation of signaling pathways into existing algorithms, in addition to providing a starting point for the development of new algorithms. However, in considering the DCA, we do not present a concrete approach to this incorporation, due to the fact that, as this work is still ongoing, there are some theoretical aspects requiring investigation before incorporating this information into the algorithm. One of these aspects is understanding to which degree the differential use of adaptors explains the emergence of specificity in the response to different TLR ligands, which can suggest the importance of additional mechanisms operating in these cells.

The need for more biologically-plausible algorithms is highlighted by Stepney et al. [5], which proposed a conceptual framework for the development of such algorithms. This framework encompasses three main steps: probing the biological system, formulating a model incorporating some of the features of the biological system, and, after validation, developing an algorithm. In turn, this is an iterative process, because each step is amended to refinements. In particular, the intermediate step involving the development of models is particularly important, as it can support the development of algorithms involving simplified models of signaling pathways (e.g. where certain molecular species are neglected) while, at the same time, allowing for a reasonable reproduction of the properties of a given pathway. In addition, when dealing with complex systems (such as signaling pathways), whose emergent behavior cannot be easily predicted from simply looking at the biological system, the importance of formulating models is further highlighted.

In fact, the development of biologically-plausible algorithms is a growing theme in AISs. Twycross and Aickelin [27] discuss the possibility of using models inspired on the immune systems of plants and invertebrates, which are relatively simpler than those of vertebrates, and the need to consider systemic models, as most real-world applications require systems based on a holistic view of the immune system. The latter is, in fact, receiving an increasing focus, especially by researchers working with homeostasis-inspired systems [28]. Finally, Guzella et al. [29] point out some signal processing capabilities of T cells, and discuss the incorporating of some of these mechanisms involved in a more biologically-plausible model of T cells. In particular, this is an interesting candidate for the incorporation of signaling pathways, which "implement" these signal processing capabilities. Additional discussions on recent developments on new immune-inspired algorithms and inspirations can be found in recent reviews and position papers [7,30,31].

In the following, we consider the DCA in greater details, although most of the discussion applies also to the TLR algorithm [26]. In the DCA, signal processing by DCs is incorporated in a simplified way (see chapter 4 in [8]), which can be described by the following equation:

$$\left[\Psi_{cs}(t)\ \Psi_{mt}(t)\ \Psi_{sm}(t)\right]^T = W(1 + I_f(t))\ \left[I_d(t)\ I_p(t)\ I_s(t)\right]^T \quad (1)$$

where Ψ_{cs}, Ψ_{mt} and Ψ_{sm} are the co-stimulation, mature and semi-mature output signals, I_f, I_d, I_p, I_s are the inflammatory, danger, PAMP and safe input signals, and the W matrix is constant. In particular, one of the PAMP (and also danger) signals in the natural immune system is the ligation of TLRs, a view which is incorporated in the DCA. The output signals define the state of a DC, which, through the application of a threshold function, determine if it will migrate, and its phenotype (mature or semi-mature) upon migration. The inflammatory signal is generally held constant, so that system 1 becomes a linear time-invariant dynamical system with dynamics faster than that of the input signals (so that the transient response can be neglected). However, the dynamics of cellular responses, which are mediated by signaling pathways, are neither linear or time-invariant.

As a consequence of the use of this simplified model of DCs in the DCA, it follows that DCs with different previous "experiences" process input signals in the same way, which may be undesirable in some applications. In particular, as discussed in section 4, the responses resulting from TLR ligation (which can be interpreted in the DCA as I_p and I_d) display features such as synergism (e.g. the non-additive secretion of cytokines following the combined stimulation of different TLRs) and antagonism, which can be attributed to nonlinearity and the temporal sequence of receptor signaling, respectively. For example, consider the application of the DCA in a simplified intrusion detection scenario, where danger and PAMP signals are mapped to suspicious activities, while safe signals indicate the normal operation of the network. In this case, a DC which has only received danger/PAMP signals up to a certain instant, which has accumulated evidence of suspicious activies, and should be more likely to acquire a mature phenotype upon migration, and a DC which hasn't received any signals yet, will process an incoming input signal in the same way, in terms of the output signals derived. By incorporating the dynamics of signaling pathways and the expression of related genes, it may be possible to improve the integration of input signals, so that if a danger-experienced DC receives another danger/PAMP signal it is more likely to migrate due to synergistic effect of two of time-correlated danger signals, even it its co-stimulation value is lower than the migration threshold.

To understand how some modifications could be incorporated, we follow the general, but simplified, representation of an agent presented in figure 4. In this representation, an agent is seen as an input-output mapping with an internal state. The inputs are stimuli received from the environment or other cells, while the output is the secretion of soluble factors. The internal state represents, collectively, the states of signaling pathways (e.g. the concentrations of certain molecular species), and the expression of genes, and is under continuous "update".

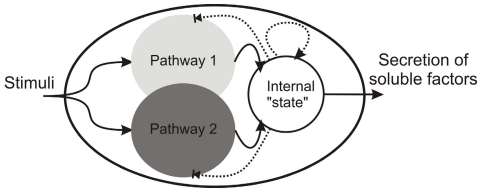


Fig. 4. A generic representation of an agent where the dynamics of signaling pathways are considered

It also influences how an input signal is received (by modulating certain signaling pathways). Upon stimulation, the internal state is modified by a response mediated by the signaling pathways, changing the state of the cell (e.g. inducing proliferation, apoptosis or other responses), in addition to the secretion of soluble factors (such as cytokines or chemokines). Therefore, the two most immediate modifications in the DCA, which would result in more biologically-plausible models of DCs, are the consideration of the transient dynamics (i.e. through the internal states), and accounting for how the internal states affect these dynamics, by defining how the internal states affect the transduction of a given signal.

This section would not be complete if the argument that, due to the inherent complexity of signaling pathways, their consideration in algorithms is not feasible at this moment, is not discussed. Although biological systems (such as the immune or nervous systems) are immensely complex, this has not precluded their use as inspiration for developing algorithms. In addition, due to growing technological advances, the biological understanding of many pathways is increasing in a fast pace, such that sufficient information on several pathways (such as those discussed in this paper) is starting to become available.

Another aspect noteworthy of discussion is an important difference between spiking neurons and signaling pathway models, which can pose some difficulties in the incorporation of the latter in computational algorithms. In contrast to models of ion channels, which can usually be described by low dimensional nonlinear systems (in the case of the FitzHugh-Nagumo [32] model, two dimensions), making their analysis and implementation relatively easy, models of signal transduction pathways may be very large, due to the usually large number of molecular species involved. Nevertheless, through an appropriate study of the key steps involved in the activation of a given pathway, it may be possible to obtain simplified models.

6 Conclusions

This paper discussed the potential of incorporating cellular signaling pathways in immune-inspired algorithms. In contrary to a simple linear, sequential, cascade, most of these pathways have a complex behavior, being capable of decision making in the face of a constantly changing environment. In the case of the immune system, these pathways lead to a response to a signal (such as receptor ligation),

initiating a quick and appropriate cellular response. We have discussed in relative detail the dynamics of Toll-like receptor signaling, which could be applied in incorporating some new aspects into the Dendritic Cell Algorithm [8,9].

While this is definitely not a simple task, it is believed that incorporating, even in a simplified way, how these pathways shape cellular responses can have an interesting impact on AISs, especially on applications that can benefit from the dynamics of these pathways. However, this incorporation should be conducted with two important aspects in mind: the need for considering the characteristics of the target problem (i.e. how these dynamics could be beneficial for some application) and the need for formulating models for studying them before applying in an algorithm. The latter is particularly important, because the emergent dynamics of signaling pathways are hard to be predicted from simply looking at the biological system and, understanding them is required, especially as emergent behavior is recently pointed out as being an evaluation criterion of immune-inspired algorithms [31].

References

1. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach, 1st edn. Springer, Heidelberg (2002)
2. Kholodenko, B.N.: Cell-signalling dynamics in time and space. Nat. Rev. Mol. Cell. Biol. 7, 165–176 (2006)
3. Voutsas, K., Adamy, J.: A biologically inspired spiking neural network for sound source lateralization. IEEE Trans. Neural Networks 18(6), 1785–1799 (2007)
4. Wang, X., Hou, Z.G., Zou, A., Tana, M., Cheng, L.: A behavior controller based on spiking neural networks for mobile robots. Neurocomputing 71, 655–666 (2008)
5. Stepney, S., Smith, R.E., Timmis, J., Tyrrell, A.M., Neal, M.J., Hone, A.N.W.: Conceptual frameworks for artificial immune systems. Int. J. of Unconv. Comp. 1(3), 315–338 (2005)
6. Freitas, A.A., Timmis, J.: Revisiting the foundations of artificial immune systems for data mining. IEEE Trans. Evol. Comput. 11(4), 521–540 (2007)
7. Timmis, J., Andrews, P., Owens, N., Clark, E.: An interdisciplinary perspective on artificial immune systems. Evolut. Intel. 1(1), 5–26 (2008)
8. Greensmith, J.: The Dendritic Cell Algorithm. PhD thesis, School of Computer Science, University of Nottingham (2007)
9. Greensmith, J., Aickelin, U., Tedesco, G.: Information fusion for anomaly detection with the dendritic cell algorithm. Inform Fusion (in press, 2008)
10. Hlavacek, W.S., Faeder, J.R., Blinov, M.L., Perelson, A.S., Goldstein, B.: The complexity of complexes in signal transduction. Biotechnol. and Bioeng. 84(7), 783–794 (2003)
11. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: From molecular to modular cell biology. Nature 402(6761) (suppl. S), C47–C52 (1999)
12. Janes, K.A., Albeck, J.G., Gaudet, S., Sorger, P.K., Lauffenburger, D.A., Yaffe, M.B.: A systems model of signaling identifies a molecular basis set for cytokine-induced apoptosis. Science 310(5754), 1646–1653 (2005)
13. Helikar, T., Konvalina, J., Heidel, J., Rogers, J.A.: Emergent decision-making in biological signal transduction networks. PNAS 105(6), 1913–1918 (2008)

14. Hlavacek, W.S., Faeder, J.R., Blinov, M.L., Posner, R.G., Hucka, M., Fontana, W.: Rules for modeling signal-transduction systems. Sci. STKE 344, re6 (2006)
15. Ciocchetta, F., Hillston, J.: Bio-PEPA: An extension of the process algebra PEPA for biochemical networks. Electron Notes in Theor. Comput. Sci. 194(3), 103–117 (2008)
16. Spicher, A., Michel, O., Cieslak, M., Giavitto, J.L., Prusinkiewicz, P.: Stochastic P systems and the simulation of biochemical processes with dynamic compartments. BioSystems 91(3), 458–472 (2008)
17. Janeway, C.A., Travers, P., Walport, M., Shlonmchik, M.: Immunobiology: the immune system in health and disease, 5th edn. Garland Publishing, Inc., New York (2002)
18. Krishna, S., Jensen, M.H., Sneppen, K.: Minimal model of spiky oscillations in NF-κB signaling. PNAS 103(29), 10840–10845 (2006)
19. Kholodenko, B.N.: Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. Eur. J. Biochem. 267(6), 1583–1588 (2000)
20. Chang, L., Karin, M.: Mammalian MAP kinase signalling cascades. Nature 410, 37–40 (2001)
21. Ashwell, J.D.: The many paths to p38 mitogen-activated protein kinase activation in the immune system. Nat. Rev. Immunol. 6, 532–540 (2006)
22. Trinchieri, G., Sher, A.: Cooperation of toll-like receptor signals in innate immune defence. Nat. Rev. Immunol. 7, 179–190 (2007)
23. Iwasaki, A., Medzhitov, R.: Toll-like receptor control of the adaptive immune responses. Nat. Immunol. 5(10), 987–995 (2004)
24. Akira, S., Takeda, K.: Toll-like receptor signalling. Nat. Rev. Immunol. 4, 499–511 (2004)
25. Whitmore, M.M., Iparraguirre, A., Kubelka, L., Weninger, W., Hai, T., Williams, B.R.G.: Negative regulation of TLR-signaling pathways by activating transcription factor-3. J. Immunol. 179, 3622–3630 (2007)
26. Twycross, J.P.: Integrated Innate and Adaptive Artificial Immune Systems Applied to Process Anomaly Detection. PhD thesis, School of Computer Science, University of Nottingham (2007)
27. Twycross, J., Aickelin, U.: Biological inspiration for artificial immune systems. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 300–311. Springer, Heidelberg (2007)
28. Owens, N.D., Timmis, J., Greensted, A.J., Tyrell, A.M.: On immune inspired homeostasis for electronic systems. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 216–227. Springer, Heidelberg (2007)
29. Guzella, T.S., Mota-Santos, T.A., Caminhas, W.M.: Towards a novel immune inspired approach to temporal anomaly detection. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 119–130. Springer, Heidelberg (2007)
30. Guzella, T.S., Mota-Santos, T.A., Caminhas, W.M.: Regulatory T cells: Inspiration for artificial immune systems. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 312–323. Springer, Heidelberg (2007)
31. Hart, E., Timmis, J.: Application areas of AIS: The past, the present and the future. Appl. Soft. Comput. 8, 191–201 (2008)
32. Fall, C., Marland, E., Wagner, J., Tyson, J. (eds.): Computational Cell Biology, 1st edn. Springer, Heidelberg (2002)

AIS Based Distributed Wireless Sensor Network for Mobile Search and Rescue Robot Tracking

Albert Ko*, Henry Y.K. Lau, and Nicole M.Y. Lee

Intelligent Systems Laboratory
The University of Hong Kong
Pokfulam Road, Hong Kong SAR
aux1496@gmail.com

Abstract. The General Suppression Control Framework (GSCF) is a framework inspired by the suppression hypothesis of the immune discrimination theory. The framework consists of five distinct components, the Affinity Evaluator, Cell Differentiator, Cell Reactor, Suppression Modulator, and the Local Environment. These reactive components, each responsible for a specific function, can generate long-term and short-term influences to other components by the use of humoral and cellular signals.This paper presents the design and application of a GSCF based distributed wireless sensor network prototyping system for tracking mobile search and rescue robots. The main objective of this physical prototyping system is to demonstrate the possibility of applying advanced Zigbee sensors to form a network that can locate a small group of mobile robots within the wireless sensor network. Another important objective of the prototyping system presented is to identify potential technological constraints in the physical system. Referencing to the result obtained, future research can be formulated and realistic simulation environment can be developed.

Keywords: Artificial Immune Systems, Humanitarian Search and Rescue, Robotics, Wireless Sensor Networks.

1 Introduction

The unprecedented number and scales of natural and human-induced disasters in the past decade has urged the emergency search and rescue community around the world to seek for newer, more effective equipment to enhance their efficiency. Tele-operated robotic search and rescue systems consist of tethered mobile robots that can navigate deep into rubbles to search for victims and to transfer critical on-site data for rescuers to evaluate at a safe spot outside of the disaster affected area has gained the interest of many emergency response institutions. Distributed wireless sensor network applied in many different fields including, medical [18], civil [17], and environment research [22], has demonstrated its value in conveying data over large area with high level of power efficiency, which is particular suitable for tracking the location of search and rescue robots in large search field.

* Corresponding author.

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 399–411, 2008.

This research demonstrates the possibility to implement GSCF, an AIS-based control framework, on a distributed wireless sensor network for tracking search and rescue robot in the open field. The goals of the research are to develop a physical prototype to demonstrate feasibility of the proposed application and to acquire realistic data to be used as simulation parameters in future search and rescue research.

This paper begins with an introduction to humanitarian search and rescue and robotics search and rescue systems. Then the paper moves on to describe the basic specifications of the wireless sensor network system. An introduction to AIS and the implementation of GSCF into the mobile robot tracking prototyping system is also included in the second half of the paper. Conclusions and future works are discussed at the end of the paper.

2 Humanitarian Search and Rescue

Over the past decade, natural and human-induced disasters claimed millions of lives and demolished astronomical sum of assets around the world. Natural disasters such as the Hurricane Marilyn in 1995 [3], the Oklahoma Tornado in 1999 [15], the Indian Ocean Earthquake [23] and Hurricane Katrina in 2005 [8], and the Pakistan Earthquake in 2005 [2], all claimed deadly and costly tolls to the affected communities. Human-induced disasters such as the civil war between Uganda government and the LRA (Lords Resistance Army) that dragged on for nearly two decades since 1987, the long-running Somali civil war since 1986, and the never-ending Palestinian conflict in Hebron and the Gaza Strip caused much more causalities than nature has ever claimed. Natural disasters usually inflict one-off damage to the community. Human-induced disasters continue to inflict damage well after the "main" conflicts have ceased. The Kosovo crisis between Albanians and Serbs as well as the crisis at Timor-Leste (formerly known as East Timor) in 1999, took place for a relatively short period of time but landmines deployed during the conflicts continue to claim lives well after the crises settled. Searching and removing landmines during and after the war can reduce civilian casualty and sooth local tension. De-mining and defusing landmines after the settlement of a war is a humanitarian responsibility that war parties should bear. However, until today, yet-cleared minefields still scatter in countries like Vietnam and Cambodia, claiming lives of ill-fated civilians.

Collapsed buildings are common field environment for humanitarian search and rescue operations. Earthquakes, typhoons, tornados, weaponry destructions, and catastrophic explosions can all generate damaged buildings in large scales. The use of heavy machinery is prohibited because they would destabilize the structure, risking the lives of rescuers and victims buried in the rubble. Only by hand should the pulverized concrete, glass, furniture and other debris be removed (see Fig. 1).

Rescue specialists use trained search dogs, cameras and listening devices to search for victims from above ground. Though search dogs are effective in finding human underground, they are as limited as human in the depth they can reach below the surface of rubbles and are unable to provide a general description of the physical environment the victim locates. Camera mounted probes can provide search specialists a visual image beyond voids that dogs cannot navigate through, however their effective range is no more than 4--6 meters along a straight line below ground surface.

Fig. 1. Pakistan earthquake 2005, locals attempting to search for survivors in a collapsed girl's college. The structure was in unstable condition; excavation and lifting machineries were prohibited from the site. (Pictures taken on site by author during mission)

3 Robot Assisted Search and Rescue Systems

Robots designed for search and rescue had been discussed in scientific literature since the early 1980's [12]; however, no actual systems had been developed or fielded until 2001. With the advancement in sensor miniaturizations and exponential increment in the speed and capability of microcontrollers, rescue robots small enough to thread through rubbles are rolling out of experimental laboratories into the catastrophic areas. The first real research on search and rescue robot began in the aftermath of the Oklahoma City bombing in 1995 [14]. Robots were not used at the bombing response, but suggestions as to how robots might have been applied were taken. In 2001, the first documented use of urban search and rescue robots took place during the 9/11 World Trade Center (WTC) disaster. Mobile robots of different sizes and capacities were deployed. These robots range from tethered to wireless operated, and from the size of a lunch box to the size of a lawnmower [21]. Their primary functions are to search for victims and to identify potential hazards for rescuers.

4 Wireless Mobile Robot Tracking System

The low-cost autonomous robotic search and rescue system presented in [11] was designed to cooperate in large quantity to search for survivors in rubbles. These robots were equipped with wireless communication module to facilitate data and video/audio transfer. These wireless robots, with no tethers, can navigate freely in obstructed environment but are difficult to track their locations once they wander out of the operators' sights. The Zigbee communication module equipped in each of these mobile robots offers an opportunity to track down their locations. The following paragraphs will describe how a Zigbee based sensor network interacts with the onboard Zigbee communication module on each robot to estimate their locations.

ZigBee (http://www.zigbee.org/en/index.asp) is a wireless technology developed to address the need for a standards-based wireless networking systems for low data-rates,

and low-power consumption applications. ZigBee supports many network topologies, including Mesh. Mesh Networking can extend the range of the network through routing, while self-healing increases the reliability of the network by re-routing a message in case of a node failure. These unique features are highly desirable for search and rescue robots operating in unstructured environment. The ZigBee-based sensor network hardware employed in this research is based on the Chipcon 2431 (http://www.ti.com/lit/gpn/cc2431) development kit (see Fig. 2).

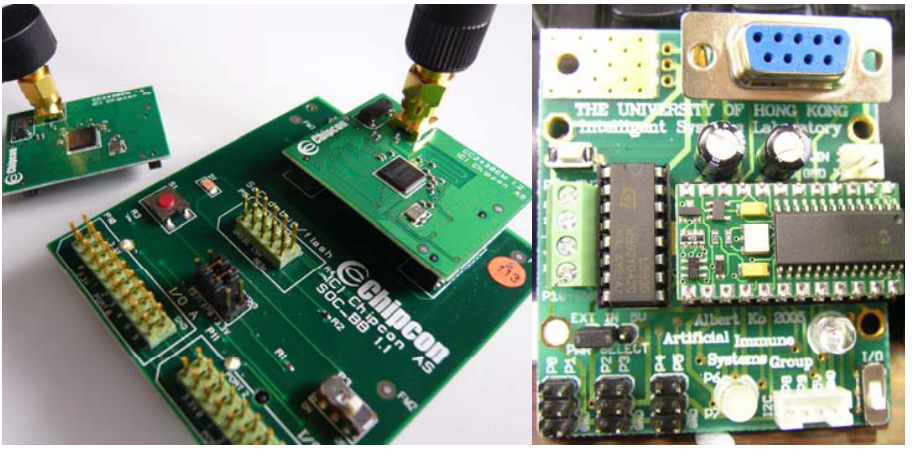


Fig. 2. Left: Two Zigbee modules, one connected to a battery-board, while the standalone one is ready to be connected to the AIS control board through serial communication. **Right:** Custom designed general purpose AIS control board installed in the two mobile robots.

The sensor network built with the 12 Zigbee modules in the development kit has 9 modules programmed as reference nodes, and 2 modules programmed as blind nodes. The 9 reference nodes were distributed around the laboratory roughly resemble a square grid as show in Fig. 3. The two blind nodes were installed on each of the two mobile robots. The last Zigbee module (or the first) of the 12 was gallantly sacrificed in short-circuit during programming.

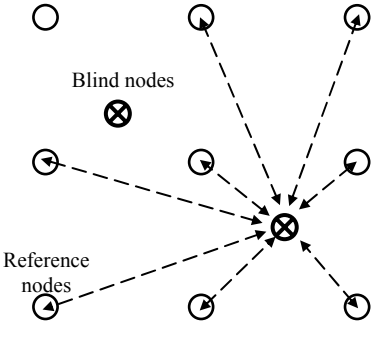


Fig. 3. Zigbee modules in grid. Reference nodes are represented by blank circles, where blind nodes are represented by crossed circles.

Reference nodes are static nodes placed at known position and can tell other nodes where they are on request. Reference nodes do not need the hardware for location detection and do not perform any calculations. Blind nodes, on the other hand, are programmed to collect signals from all reference nodes responding to their request; then read out the respective RSSI values, feed the values into the location engine, and afterwards read out the calculated position and send to the control console. Since all location calculations are performed at each blind node, the algorithm is genuinely decentralized. This property reduces the amount of data transferred in the network, since only the calculated position is transferred, not the data used to perform the calculation. The system is therefore highly scalable.

The ZigBee modules used are embedded with 8051 8-bit single-cycle processor, 128 KB in-system programmable flash, and 8 KB RAM, which adds up to roughly 8 times the performance of a standard 8051. This processing power allows the blind nodes to use up to 16 reference nodes to estimate its current position. In theory, signals from 3 reference nodes is the least to make a sensible estimation, the more reference node signals received, the more accurate the estimation is.

Algorithm used to estimate locations of the blind nodes within the sensor network is straightforward. To estimate its current location, the blind node on the mobile robot broadcast a specific signal to the surrounding. All reference nodes within range response to the signal by sending a packet containing the reference nodes' relative coordinate. The algorithm uses Received Signal Strength Indicator (RSSI) values to estimate distance from each reference node. Since RSSI value decreases as distance increases, the blind node would chose the 8 nearest reference nodes by comparing RSSI values between all reference nodes in range. Based on the strength of these returned signals and the origin of each signal included in the packet, position of the blind node can be estimated.

5 Biological and Artificial Immune Systems

Human immune system is a robust, efficient, and adaptive system. The immune system continuously acquires new knowledge of non-self cells, adjusts its responses against foreign antigens, scales up defense mechanism to foil foreign attacks, suppresses destructive actions against self cells, converts emergent behaviors into organized memories, and stores distributed memories for global access. Artificial Immune Systems (AIS) [7] is a new computational intelligence paradigm built around inspirations from its biological counterpart. This new computational paradigm, in general, focuses to exploit and mimic the four main functions in the biological immune system by embedding various computational techniques and algorithms. These artificial functions are further integrated to form decentralized systems with specific advantages to meet application needs. Many of these systems had successfully implemented to decentralized systems to perform learning, data manipulation, abnormality detection, object classification and pattern matching.

Scientists and engineers have applied AIS to solve a wide variety of problem. [6] presented the application of AIS in computer network security, machine learning, and pattern recognition. [19] applied attribute weighted AIS to diagnosis heart and diabetes diseases. [5] exploited negative selection algorithm to detect abnormalities in aircrafts. [4] developed an AIS real-time visual analysis system for surveillance based on the behavior of T-cells. [16] developed AIS for detecting junk e-mail and achieved accuracy

close to and even exceeded commercial products in certain aspects. In an effort to develop robust and decentralized control systems for modular robots, [9] developed a General Suppression Control Framework (GSCF) for designing control systems for modular robots based on the suppression mechanism in AIS. This paper, continuing from previous works, describes the application of GSCF in developing decentralized wireless sensor network for tracking a small platoon of search and rescue robots.

6 General Suppression Control Framework

The General Suppression Control Framework (GSCF) is based around the analogy of the immunological suppression hypothesis in the discrimination theory [1]. The major recognition and reaction functions of the acquired immunological response are performed by T-lymphocytes (T-cells) and B-lymphocytes (B-cells) which exhibit specificity towards antigen. B-cells synthesize and secrete into the bloodstream antibodies with specificity against the antigen, the process is termed Humoral Immunity. The T-cells do not make antibodies but seek out the invader to kill; they also help B-cells to make antibodies and activate macrophages to consume foreign matters. Acquired immunity facilitated by T-cells is called Cellular Immunity.

When a T-cell receptor binds to a peptide with high affinity presented by an APC (Antigen Presenting Cells), such as macrophages, the T-cell recognized the antigen become mature and it has to decide whether to attack the antigen aggressively or to tolerate it in peace. An important decision factor is the local environment within which the T-cell resides. The present of inflammatory cytokine molecules such as interferon-gamma (INF-γ) [20] in the environment tend to elicit aggressive behaviors of T-cells, whereas the anti-inflammatory cytokines like IL-4 and IL-10 tend to suppress such behavior by blocking the signaling of aggression. In brief, a T-cell matured after recognizing an antigen does not start killing unless the environment also

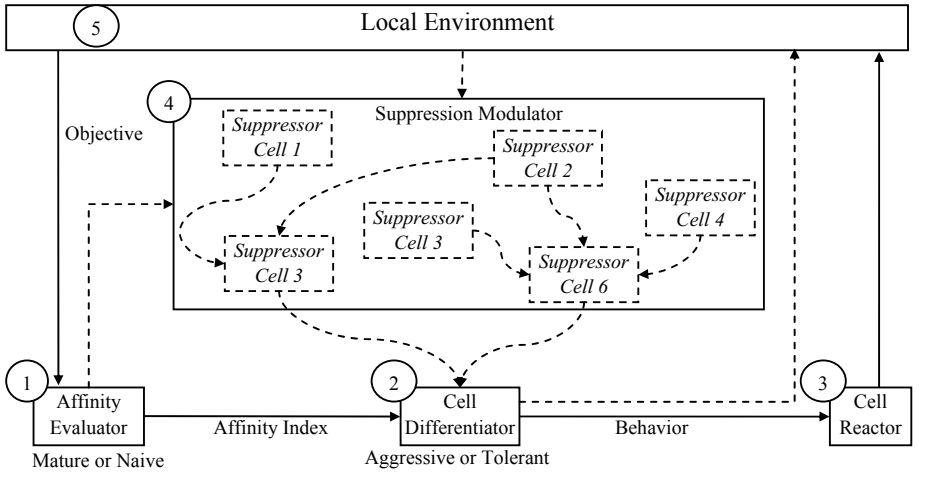


Fig. 4. The General Suppression Control Framework. Dashed lines represent humoral signal transmissions, where solid lines represent cellular signals. The suppression modulator can host any number of suppressor cells.

contains encouraging factors for doing so. In addition, after a mature T-cell developed the behavior, it will emit humoral signals that have slower transmission speed but longer lasting effect than cellular signals to convert others to join.

Our analogy infers each module of the modular robot is an autonomous T-cell that continuously reacts to the changing environment and affects the functioning of other cells through the environment. The framework consists of five major components. The most notable mechanism shown in Fig. 4 is that the T-cell's functions are divided into three separate components, the Affinity Evaluator, Cell Differentiator and the Cell Reactor. Delegating the three unique functions into separate components enables the system to be organized in a modular manner and that when programming for an application, the result and effect of each component can be observed easier. There are five main components in GSCF; they are Affinity Evaluator, Cell Differentiator, Cell Reactor, Suppression Modulator, and the Local Environment. Their functions are explained below.

Affinity Evaluator – evaluates information in the *Local Environment* against the objective and output an affinity index.

Cell Differentiator – evaluates inputs from the *Affinity Evaluator* and *Suppression Modulator* to determine the type of behavior to react.

Cell Reactor – reacts to the cellular signal from the *Cell Differentiator* and executes the corresponding behaviors that take effect in the *Local Environment*.

Suppression Modulator – is a collection of *Suppressor Cells* that are sensitive to predefined external stimulants.

Local Environment – is where interactions between different components take place and a theoretical space to integrate the physical objects and the abstract system in an analyzable form.

7 Distributed Wireless Sensor Network

The distributed wireless robot tracking system presenting in this paper is based on the GSCF [10] developed for controlling decentralized systems. To design a GSCF based system, system objective and system constraints must first be identified. For the wireless robot tracking system in this research, the primary objective is to continuously track the location of each robot by evaluating a collective set of feedbacks from multiple sources. These feedbacks include coordinates from the Zigbee Communication Module, motor encoders, and electronic compass. The only system constraint to be incorporated into the system is accuracy of the estimated robot locations.

The low-cost Zigbee based sensor network used in this research is suitable for tracking robots in large area and to relate information over long distance in an energy efficient manner. However, position estimations obtained from RF based systems are venerable to interferences; therefore additional referencing sensors are often desirable in more accurate applications. The solution for this particular application is to take advantage of the readily available motor encoders and electronic compasses installed in the robots to generate more reliable position estimations, though these sensors all exhibits inherited reliability issues in their own way. Table 1 lists their advantages and disadvantages.

Table 1. Advantages and disadvantages of the three feedbacks used in the system

Sensor types	Advantages	Disadvantages
Received Signal Strength Indicator (RSSI)	Covers large area Low Power Consumption	Venerable to interferences
Electronic Compasses	High accuracy	Slow response time
Motor Encoders	High precision	Cannot detect slippage

Based on the strength and weaknesses of each type of sensors listed above, RSSI is a more reliable source to estimate the robots position fast without accumulative error. Motor encoders are not reliable for long distance tracking as slippage error would accumulate, however it is good for short distance position tracking. Electronic compasses can be used to confirm the direction in which the robot is moving towards, which in turn can verify the accuracy of the coordinates produced using RSSI estimation. In general, the blind node on the mobile robot would sample surrounding reference nodes 10 times per estimation. Each set of 10 RSSI returned per reference node are converted to distances. The highest and lowest readings are removed from the sample set, and then standard deviation of the remaining readings in the data set is produced to evaluate the reliability of the estimated distance. The estimated distance is more reliable if the standard deviation is low, otherwise the reliability is low.

For GSCF, the fundamental idea is to let Affinity Evaluator to decide whether there is a problem to solve (an system objective to pursue), and then consult the Cell Differentiator to decide whether the system has the resources to solve the problem under imposed constraints. For the distributed wireless robot tracking system presented, the Affinity Evaluator is responsible for monitoring the status of the system objective. The system objective is said to have achieved when the estimated distance is reliable (i.e. standard deviation of RSSI is low). The Affinity Evaluator would produce a low affinity index when the system object is achieved to encourage the system to behave in tolerant mode. When the system is in tolerant mode, it would rely on the primary data source (RSSI) to estimate robot positions. Otherwise, the Affinity Evaluator would produce a high affinity index to alert the system abnormality is detected and lead the system to enter aggressive mode. Under aggressive mode, the system would use additional sensing sources, encoders and compasses, to make better estimation. While the affinity index is high, Cell Differentiator would actively evaluate various system constraints to see how the robot should behave. These constraints being evaluated may be predefined system constraints or newly developed constraints due to changes in the environment. GSCF define these constraints as suppressor cells (SC), these cells may evolve to adapt to new changes and may proliferate to increase their sensitivity to specific stimulants.

The distributed wireless robot tracking system under discussion has two additional sensor sources that influence the robots' behaviors. The encoder tells the displacement of the robot by counting rotations made by the motor. The electronic compasses read the robots direction at any instant with reference to the earth's magnetic field.

Suppressor cells that have high sensitivity to the changes of these sensors readings are situated in the Suppression Modulator. The following paragraph discusses how suppressor cells are designed and how individual suppressive action can be combined to produce useful results. Though there are only three types of sensor sources, there are six types of suppressor cells in the system. Table 2 lists their functions.

Suppression Modulator is a very important component in GSCF; it contains suppressor cells that are sensitive to particular sensors and can be viewed as representations of external constraints reacting inside the control system. Functions of the first four suppressor cells listed in Table 2 are self-explanatory. The function of summation cell SC5 is designed to compare the estimated traveled distance from encoder and from the sensor network. For example, a mobile robot driving against an obstacle would report high counts on the encoder but the estimated position reporting from the sensor network would probably remain unchanged. This discrepancy between estimations from two sensors would reflect in the suppression index produced by SC5, the higher the discrepancy level, the higher the suppression index (see illustration in Fig. 5). Function of summation cell, SC6, is similar to that of SC5, except it considers an additional constraint. SC6 determines whether the readings obtained from sensor network is reliable by comparing the estimated direction from sensor network against the reading from electronic compass. SC6 takes in the initial and final estimated locations from sensor network to trigonometrically estimate the direction the robot is moving, then compare this estimation against the electronic compass reading from SC3 to produce a suppression index that reflects the discrepancy, the higher the discrepancy level, the higher the suppression index. Suppression index from SC5 and SC6 are crucial for Cell Differentiator to adapt a behavior that best fit the situation.

Table 2. Summary of suppressor cells in the *Suppression Modulator*

	Sensitive Element	**Suppressor Cell Duties**	**Output to Cell Differentiator**
SC1	Encoders	Estimate traveled distance in respect to encoder readings	Output to SC5
SC2	Sensor Network	Estimate traveled distance in respect to RSSI readings	Output to SC5 Output to SC6
SC3	Electronic Compass	Estimate traveling direction in respect to compass readings	Output to SC6
SC4	Sensor Network	Estimate traveling direction in respect to RSSI readings	Output to SC6
SC5	Summation Cell (SC1 and SC2)	Combines signals from SC1 and SC2 to produce a suppression index representing the compliance of readings from SC1 and SC2 Suppression index lowest when the two readings agree.	Suppression Signal 1-10
SC6	Summation Cell (SC2, SC3, SC4)	Combines signals from SC2, SC3 and SC4 to produce a suppression index that represent the compliance of readings from SC2 and SC3 in respect to condition of SC2 Suppression index lowest when the two readings agree.	Suppression Signal 1-10

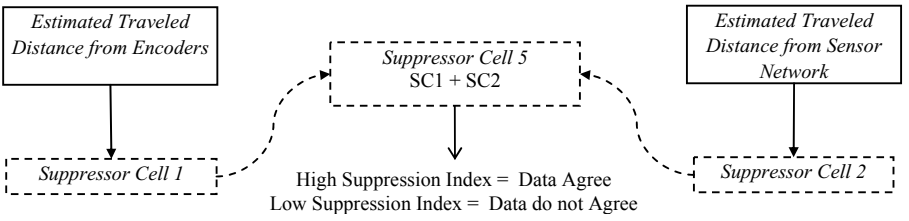


Fig. 5. The function of SC1, SC2, and SC5 illustrated as an independent system. In short, SC5 fuses data for *Cell Differentiator* to evaluate.

The function of Cell Differentiator is similar to the biological cell differentiation mechanism, in which cells develop aggressive or tolerant behavior in response to the type of cytokines present in the immune system. Cell Differentiator is responsible for integrating complex information from different sources into simple instructions and converts intricate problems into quantitative outputs. The decision flow of the Cell Differentiator can be summarized in a flow chart as shown in Fig. 6.

The suppression indices from the suppressor cells have priority over all others, it is being evaluated first to see whether the estimation based on encoders, sensor network, and compasses comply with each other. If the suppression index is low, meaning the estimation from sensor network agree with additional sources (encoder and compass); the suppressor modulator will not react strongly. If the Affinity Index is low, meaning the RSSI data is sable, the system will behave in tolerant mode. Otherwise, the suppression index is high or the affinity index is high, the system will switch into aggressive mode to

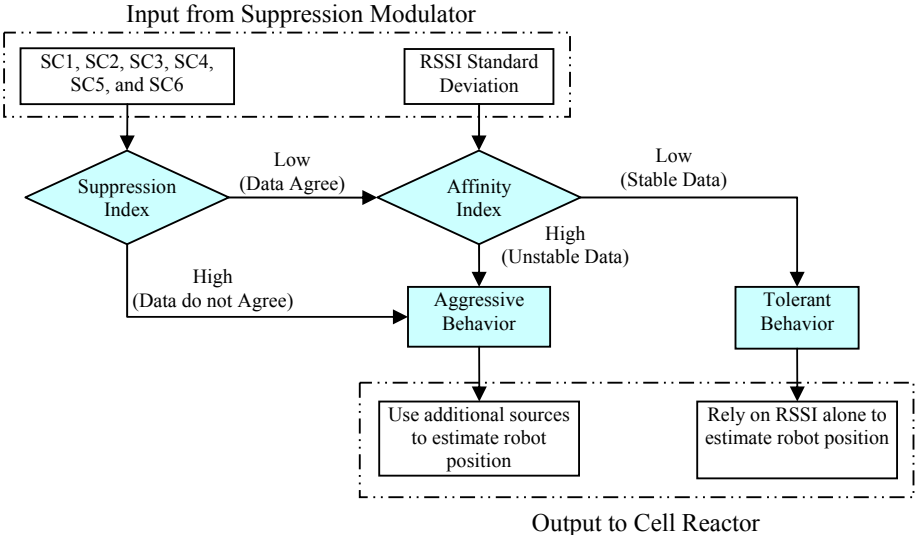


Fig. 6. Decision scheme in the Cell Differentiator of each modular fireguard

Since the Cell Differentiator in GSCF is only responsible for producing high-level behavioral instructions such as “sound the alarm”, “stand fast”, “search for heat”, etc. There has to be a component to interpret these high level commands into lower level commands for the mechanical controllers. This component is called Cell Reactor. Since mechanical control schemes varies greatly between different operation platforms, GSCF delegates this work to Cell Reactor, so the high level design of other components can remain platform independent.

8 Conclusions

The AIS-based distributed tracking system developed for the mobile search and rescue robots is being tested indoor in a laboratory between tables, chairs and miscellaneous obstacles. Within the environment there are uncontrolled RF interferences of different sorts, including Wi-Fi routers, mobile phones, activated RFID systems, Bluetooth devices (keyboard and mouse), and EMF from various mechanical devices. Despite the abundant sources of interferences, the test environment is far from practical for what this system is designed for.

For search and rescue robots to be able to carryout practical tasks in the field, the tracking system must be equipped with the ability to trace robots’ positions even when it navigated underneath the rubble. To track robots underneath rubble would probably require mobile robots to form a mobile tracking network in addition to the stationary tracking network. Limited number of Zigbee modules in hand is one limiting factor to carryout the experiment in condition closer to field environment. Designing a method to evaluate accuracy of position estimation after the robot navigates and burry itself deep into rubble is another difficult issue. These will be works to follow.

Since this is a prototype for tracking and controlling low-cost autonomous search and rescue robots with GSCF based sensor network; demonstrating the system’s functionality and recording realistic data from the physical system to implement into future simulations is the main goal at this stage. In general, the performance of the robot is inline with design expectation and the GSCF based tracking system works well as the backbone of the system. Data from the experiment cannot be extracted from the system due to technical difficulties; detail will be discussed in future works. To further develop and evaluate the current prototype system, points of improvements are discussed in the following session.

9 Future Works

The prototype presented in this paper relies heavily on the original interface came with the development kit, which limits the programming and real-time data output capacity of the modules. Immediate work is to develop a programming and data manipulations interface that for extracting data from the system for graphical and mathematical analysis.

Long term work is to develop methods to evaluate accuracy of sensor network estimated position against actual position in obstructed environment, i.e. in rubble. This work would provide a base to compare and evaluate results of different control and

tracking algorithms. In addition, technologies and methods that can help to setup the system quickly for emergency application is another important area to make the system truly applicable.

Acknowledgement

The work described in this paper was partly supported by the Research Grant Council of the Hong Kong Special Administrative Region, PRC under the CERG Project No. HKU7142/06E.

References

1. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., Mcleod, J.: Danger Theory: The Link Between AIS and IDS. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
2. Birsel, R.: Pakistan: Frantic Search as Pakistani Quake Toll Tops 20,000. ReliefWeb (October 10, 2005), http://www.library.cornell.edu/newhelp/res_strategy/citing/apa.html#apa (retrieved June 12, 2006)
3. Centers for Disease Control and Prevention: Deaths Associated With Hurricanes Marilyn and Opal – United States, September-October 1995 (January 19, 1996), http://www.cdc.gov/mmwr/preview/mmwrhtml/00040000.htm (retrieved June 27, 2006)
4. Cserey, G., Porod, W., Roska, T.: An Artificial Immune System based Visual Analysis Model and Its Real-Time Terrain Surveillance Application. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 250–262. Springer, Heidelberg (2004)
5. Dasgupta, D., KrishnaKumar, K., Wong, D., Berry, M.: Negative Selection algorithm for Aircraft Fault Detection. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 1–13. Springer, Heidelberg (2004)
6. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
7. de Castro, L.N., Timmis, J.: Artificial Immune Systems as a Novel Soft Computing Paradigm. Softing Computing 7(8), 526–544 (2003)
8. Federal Emergency Management Agency: Cash Sought to Help Hurricane Victims, Volunteers Should Not Self-Dispatch (August 29, 2005), http://www.fema.gov/news/newsrelease.fema?id=18473 (retrieved June 27, 2006)
9. Ko, W.Y.A., Lau, H.Y.K., Lau, T.L.: A decentralized control framework for modular robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), Sendai, Japan, pp. 1774–1779 (2004a)
10. Ko, A., Lau, H.Y.K., Lau, T.L.: An immuno control framework for decentralized mechatronic control. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 91–105. Springer, Heidelberg (2004b)
11. Ko, A., Lau, H.Y.K., Lau, T.L.: An Immuno Control Framework for Decentralized Mechatronic Control. International Journal of Unconventional Computing - 2005 Special Issues, 225–280 (to appear, 2005a)
12. Kobayashi, A., Nakamura, K.: Rescue Robot for Fire Hazards. In: Proc. of International Conference on Advanced Robotics, pp. 91–98 (1983)

13. Lau, Y.K.H., Wong, W.K.V.: Immunologic responses manipulation of AIS agents handling. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 65–79. Springer, Heidelberg (2004)
14. Murphy, R.R.: National Science Foundation Summer Field Institute for rescue robots for research and response (R4). AI Magazine 25(2), 133–136 (2004a)
15. National Severe Storms Laboratory: Oklahoma/Kansas Tornado Outbreak (May 3, 1999), http://www.nssl.noaa.gov/headlines/outbreak.shtml (retrieved May 22, 2006)
16. Oda, T., White, T.: Immunity from Spam: An analysis of an Artificial Immune System for Junk Email Detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 276–289. Springer, Heidelberg (2005)
17. Paek, J., Chintalapudi, K., Govindan, R., Caffrey, J., Masri, S.: A Wireless Sensor Network for Structural Health Monitoring: Performance and Experience. In: Proc. 2nd IEEE Workshop on Embedded Networked Sensors. IEEE CS Press, Los Alamitos (2005), http://www.cse.unsw.edu.au/~emnet
18. Patel, S., Lorincz, K., Hughes, R., Huggins, N., Growdon, J.H., Welsh, M., Bonato, P.: Analysis of Feature Space for Monitoring Persons with Parkinson's Disease With Application to a Wireless Wearable Sensor System. In: Proceedings of the 29th IEEE EMBS Annual International Conference, Lyon, France (August 2007), http://www.eecs.harvard.edu/~mdw/papers/parkinsons-embs07.pdf
19. Sahan, S., Polat, K., Kodaz, H., Gunes, S.: The Medical Applications of Attribute Weighted Artificial Immune System (AWAIS): Diagnosis of Heart and Diabetes Diseases. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 65–79. Springer, Heidelberg (2005)
20. Sharon, J.: Basic Immunology. Pennsylvania. Williams & Wilkins, Pennsylvania (1998)
21. Snyder, R.: Robots assist in search and rescue efforts at WTC. IEEE Robot. Automation Magazine 8, 26–28 (2001)
22. Werner-Allen, G., Lorincz, K., Welsh, M.: Deploying a Wireless Sensor Network on an Active Volcano. IEEE Internet Computing, 18–25 (March 2006)
23. Zubair, L.: Scientific Background on the Indian Ocean Earthquake and Tsunami, The International research Institute for Climate and Society (December 28, 2004), http://www.library.cornell.edu/newhelp/res_strategy/citing/apa.html#apa (retrieved June 27, 2006)

Eating Data Is Good for Your Immune System: An Artificial Metabolism for Data Clustering Using Systemic Computation

Erwan Le Martelot[1,3], Peter J. Bentley[2], and R. Beau Lotto[3]

[1] Engineering Department, University College London, London, UK
e.le_martelot@ucl.ac.uk
[2] Computer Science Department, University College London, London, UK
[3] Institute of Ophthalmology, University College London, London, UK

Abstract. Previous work suggests that innate immunity and representations of tissue can be useful when combined with artificial immune systems. Here we provide a new implementation of tissue for AIS using systemic computation, a new model of computation and corresponding computer architecture based on a systemics world-view and supplemented by the incorporation of natural characteristics. We show using systemic computation how to create an artificial organism, a program with metabolism that eats data, expels waste, clusters cells based on the nature of its food and emits danger signals suitable for an artificial immune system. The implementation is tested by application to a standard machine learning set and shows excellent abilities to recognise anomalies in its diet.

1 Introduction

An increasingly popular view in the field of Artificial Immune Systems (AIS) holds that innate immunity (as enabled by non-adaptive cells such as dendritic cells) can play a significant role in maintaining immunity in computer systems [1]. Notions such as the Danger Theory suggest that normal self cells may provide signals when damaged, thus helping to encourage the response of immune cells in the right areas of the tissue of an organism at the right time [2]. Previous work has investigated the development of an artificial tissue to serve this function, providing an interface between data and AIS, and performing preliminary data processing and clustering [3].

In this work we extend the previous work on tissue for AIS, and investigate a different implementation based on the recent paradigm and computer architecture, systemic computation (SC) [4] designed to support any bio-inspired system by enabling natural characteristics found in biology. In contrast to previous implementations of tissue, which largely ignore the relationships between real organisms and their environments, here we present a model of organism, implemented as a systemic computation program with its own metabolism that eats data, expels waste, clusters its cells depending on the nature of its food and can emit danger signals for an AIS. The implementation is tested by application

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 412–423, 2008.

to a standard machine learning set (Breast Cancer data [5]) and shows excellent abilities to recognise anomalies in its diet.

2 Background

Although not commonly modelled, the notion of tissue is fundamental to immunity. The immune system within an organism defends the tissue of that organism. The concept of artificial tissue has been used for instance in the POEtic project, aiming at creating a hardware platform organised with a similar hierarchy as found in biological systems [6], and using reconfigurable circuits to simulate tissue growth [7]. It has also been used in work that implemented an AIS in a sensor network, the sensor nodes taking on the role of tissue cells [8].

In biology, tissue is a crucial part of the immune system and its importance was particularly highlighted by Polly Matzinger when introducing the Danger Model [2]. This view rejected the notion that the immune system differentiates self from non-self and suggested that it instead responds to cellular damage. It thus suggests that cells that die abnormally release signals which encourage immune cells to converge on that location and become more active.

This theory was adopted in [3] to propose two ways of growing tissues where damaged cells would release danger signals exploitable by an AIS. Tissue was defined as the interface between a problem to solve and the AIS. Here we follow a similar view, but attempt to improve the tissue model and its potential advantages by implementing a tissue-growing program designed for AIS using systemic computation - a parallel computer architecture designed to support natural computation.

Systemic computation is not the only model of computation to emerge from studies of biology. The potential of biology had been discussed in the late 1940s by Von Neumann who dedicated some of his final work to automata and self-replicating machines [9]. Cellular automata have proven themselves to be a valuable approach to emergent, distributed computation [10]. Generalisations such as constrained generating procedures and collision-based computing provide new ways to design and analyse emergent computational phenomena [11] [12]. Bio-inspired grammars and algorithms introduced notions of homeostasis (for example in artificial immune systems), fault-tolerance (as seen in embryonic hardware) and parallel stochastic learning, (for example in swarm intelligence and genetic algorithms) [4].

New architectures are also popular, whether distributed computing (or multi-processing), computer clustering and grid computing and even ubiquitous computing and speckled computing [13]. Thus, computation is increasingly becoming more parallel, decentralised and distributed. However, while hugely complex computational systems will be soon feasible, their organisation and management is still the subject of research. Ubiquitous computing may enable computation anywhere, and bio-inspired models may enable improved capabilities such as reliability and fault-tolerance, but there has been no coherent architecture that combines both technologies. Indeed, these technologies appear incompatible - the computational

overhead of most bio-inspired methods is prohibitive for the limited capabilities of ubiquitous devices.

To unify notions of biological computation and electronic computation, [4] introduced systemic computation as a suggestion of necessary features for a computer architecture compatible with current processors, yet designed to provide native support for common characteristics of biological processes.

In this paper we use an approach similar to [3] and deepen the biological analogy by modelling an artificial organism as a program with metabolism. The program does not only mimic some tissue features but also mimics many fundamental properties of living organisms: eating data as food and expelling waste, while growing tissue, and releasing danger signal when its cells die in an abnormal way.

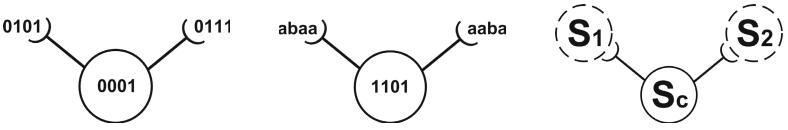
To implement such program SC provides a suitable alternative approach to traditional computation. Indeed with SC, organisms and software programs now share a common definition of computation. The work illustrates how organisms and programs can behave similarly, sharing the notion of metabolism, using SC.

3 Overview of Systemic Computation

SC [4] is a new model of computation and corresponding computer architecture based on a systemics world-view and supplemented by the incorporation of natural characteristics (previously listed). This approach stresses the importance of structure and interaction, supplementing traditional reductionist analysis with the recognition that circular causality, embodiment in environments and emergence of hierarchical organisations all play vital roles in natural systems. Systemic computation makes the following assertions:

- Everything is a system.
- Systems can be transformed but never destroyed.
- Systems may comprise or share other nested systems.
- Systems interact, and interaction between systems may cause transformation of those systems, where the nature of that transformation is determined by a contextual system.
- All systems can potentially act as context and affect the interactions of other systems, and all systems can potentially interact in some context.
- The transformation of systems is constrained by the scope of systems, and systems may have partial membership within the scope of a system.
- Computation is transformation.

In systemic computation, everything is a system, and computations arise from interactions between systems. Two systems can interact in the context of a third system. All systems can potentially act as contexts to determine the effect of interacting systems. A system is divided into three parts: two schemata and one kernel. These three parts can be used to hold anything (data, typing, etc.) in binary as shown in Figure 1(a). The kernel defines the result of two systems interacting in its context (and may also optionally hold data if it is interacting with



(a) Data system (b) Context system (c) Interaction

Fig. 1. 1(a): A system used primarily for data storage. The kernel (in the circle) and the two schemata (at the end of the two arms) hold data. 1(b): A system acting as a context. Its kernel defines the result of the interaction while its schemata define allowable interacting systems. 1(c): An interacting context. The contextual system S_c matches two appropriate systems S_1 and S_2 with its schemata and specifies the transformation resulting from their interaction as defined in its kernel.

another system). The two schemata define which subject systems may interact in this context as shown in Figures 1(b) and 1(c). A system can also contain or be contained by other systems. This enables the notion of scope. Interactions can only occur between systems within the same scope. An SC program therefore comprises systems that are instantiated and positioned within a hierarchy (some inside each other). It thus defines an initial state from which the systems can then randomly interact, transforming each other through those interactions and following an emergent process rather than a deterministic algorithm. For full details see [4] and [14].

Systemic Computation has been used to model genetic algorithms, neural networks, and has demonstrated properties of flexibility, fault tolerance, and self-repair [14], [15], [16].

4 An SC Program with Metabolism

4.1 Systemic Analysis

When programming with SC it is necessary to perform a systemic analysis in order to identify and interpret appropriate systems and their organisation [14]. The first stage is to identify the low-level systems (i.e. determine the level of abstraction to be used).

In most artificial immune systems, the level of abstraction is the cell: few approaches require modelling of the internal organelles or genome of cells, and few require modelling of populations of organisms. Here we intend to model the growth of tissue cells, the consumption of "food" (data items), the expulsion of waste and the emission of danger signals. Thus an abstraction at the cellular level is appropriate, with systems being used to explicitly model each element.

The identification of appropriate low-level systems is aided by an analysis of interactions. The organism should be able to eat food from its environment, use this food to grow organs (clusters of cells) by creating new cells and expel waste into the environment.

To prevent being overloaded with systems, the waste can be recycled into new food (a simple ecosystem model). Food and waste could therefore be seen as

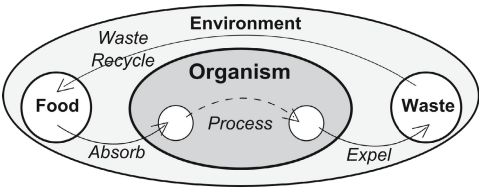


Fig. 2. 'Food to waste' cycle for an organism within its environment: Food is absorbed by the organism, processed as energy to grow tissues before being expelled when the organism cannot make use of it any more

different states of the same system (in SC systems can be transformed, but never created from nothing or destroyed). Also, the food is what the organism takes from its environment to be able to grow. Therefore cells and all the necessary matter for the growth should also derive from the food systems.

We can thus visualise the ecosystem between the organism and the environment as shown in Figure 2.

Looking within the organism, it takes food as input and this food must be sufficient to grow tissue. One simple way to model this is by using the approximation that the food is transformed into cells when absorbed by the organism. However, to enable cells to adhere to each other (rather than float free), cells need some sticky adhesion molecules. Here we do not need to explicitly model all these molecules but an "adhesion surface" is at least required to bind two or more cells together. As SC forbids the creation of systems from nothing, the adhesion surfaces must be obtained either from incoming food or from the cells themselves. In a biological organism each cell has a limited lifespan and thus dies at some point. It may then be consumed by macrophages or dendritic cells and its energy is partially recycled. In the model dead cells can thus be recycled to make adhesion surfaces. A growth process can now attach cells to each other by using adhesion surfaces to create tissue. To regulate this growth and introduce the notion of time, a decay process simulates the aging of cells. When cells die, a split process splits them from the adhesion surfaces they are bound to.

So the organism eats new data, converts each data item into a new cell, and attempts to bind that cell to itself, with cells made from similar data items binding to each other. Thus, a cell unable to bind to any group of cells reveals itself to be significantly different from them - more like the result of an invading pathogen then part of the organism. If this abnormal cell dies unbound, it can therefore be spotted as a potential anomaly. In that case, the death of the cell can entail that cell releasing a Danger signal (i.e. the cell can be converted into a signal). This signal can then be used by an AIS algorithm which can be implemented through the addition of systems corresponding to immune cells. (Here we focus on the organism.)

The organism can also make use of a hunger parameter defining a maximum amount of alive cells it can contain at a time. This parameter can be stored in the organism system and the absorption context then only allows food absorption if the organism is "hungry". This parameter can be useful to avoid having

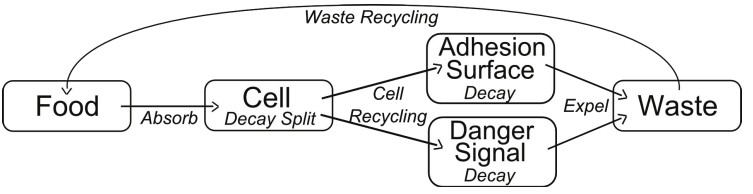


Fig. 3. 'Food to Waste' cycle within the organism: Food is absorbed, transformed into cells. When dying cells can be recycled into adhesion surfaces if they were part of a tissue or turned into a danger signal if they were single. Cells, adhesion surfaces and danger signals have a limited lifespan and decay over time (i.e. when they reach a certain age they die). When dying, cells also need a split process to detach them from the tissue they were part of.

the organism growing too big and using too much memory/data at a time. A bad usage of memory could indeed to some extend slow down the computation process significantly.

The organism food to waste chain is therefore as shown in Figure 3.

From this defined cycle, the interactions and systems in the model can be written as follows (also see Figure 4):

organism }-absorb-{ food	→ organism(cell)
cell }-growth-{ adhesion_surface	→ cell(adhesion_surface)
cell(adhesion_surface) }}-split	→ (cell adhesion_surface)
organism(cell) }}-cell_recycling	→ organism(adhesion_surface or danger_signal)
X[age](time) }}-decay	→ X[age+1](time),X=cell or adhesion_surface or danger_signal
organism(X) }}-expel	→ (organism waste), X=adhesion_surface or danger_signal
universe(waste) }}-waste_recycling	→ universe(food[data])

The absorb system models endocytosis (e.g. via cell receptors), the growth system models the organism's genome, the decay models the aging (progression along the axis of time), the split system models a chemical breakdown between adhesion molecules and cell wall, the cell recycling models the phagocytes, the expel system models exocytosis, waste recycling systems model the ecosystem, the universe models the environment, the organism system models the boundary between tissue and environment, food systems model nutrients, cells model tissue cells, adhesion surfaces model adhesion molecules, danger signal systems model Matzinger's danger signals, waste systems model cell waste (unused or unusable compounds), and the time system models the dimension of time.

Figure 4 summarises the organism's organisation and shows the potential interactions.

Note that waste recycling, absorption, cell recycling and expel systems should have the same amount of instances. Indeed, on average if one food system can be created at a time, then only one can be absorbed, then recycled and finally expelled at a time.

4.2 Data Clustering within the Artificial Organism

So far an organism has been modelled within SC. To use this organism for data clustering the data processing method has to be defined.

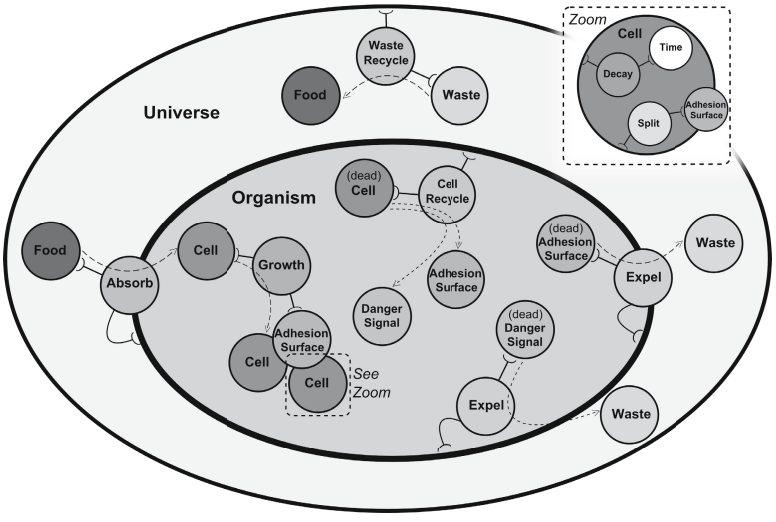


Fig. 4. Systemic organisation of the organism. The universe contains a waste recycling system, some waste and food, and an organism. The organism shares with the universe the absorption context. It contains cells, adhesion surfaces, danger signals, growth contexts, cells recycling contexts and expelling contexts. Finally cells (and thus all derived system states like adhesion surfaces and danger signals) contain the time system, a decay process and a split process. The schemata appear on context systems to show the allowable interactions between systems. The dashed arrows indicate the potential transformation of some systems during an interaction. For instance on the far left we can observe a food system interacting with an organism in an absorption context: the food is turned into a cell and injected into the organism.

To incorporate data into the organism's metabolism, new data items are placed into food systems, where it is stored in the schemata. Data from an incoming stream can be introduced when recycling waste (i.e. new data are pushed into the resulting food systems). The amount of waste recycling and absorption systems gives the data introduction rate (the more food can be absorbed at a time, the more data are introduced). The data are then absorbed into the organism and transformed to cells. When a growth interaction occurs between a cell and an adhesion surface, the two are bound based on their data similarity. Algorithm 1 describes in pseudo-code the binding method. For binding a cell to an adhesion surface the adhesion surface is injected into the cell but remains also part of the organism so that more cells can bind to it.

The measure chosen in this implementation to compare data is the Euclidian distance (as was used in [3]). In the organism, cells cluster according to their values, and various clusters may emerge from this, thus reflecting the data distribution. If a cell is left single then it means it cannot bind anywhere and therefore holds data significantly different from the current most common data values contained within the organism. This cell is then turned into a danger signal holding the data that an AIS could use to develop antibodies.

Algorithm 1. Pseudo-code for the growth context binding method. τ is a given threshold. The distance function calculates the Euclidian distance of two vectors.

```
if adhesion surface not bound to anything then
  Bind cell and surface
  Surface data value ← Cell data value
else if distance(Cell data, Surface data) ≤ τ then
  Bind cell to surface
  Surface data value ← Average(Surface data, Cell data)
end if
```

5 Experiments and Results

To test the model and compare it with similar previous models [3], a series of experiments were performed using the standard "breast cancer" UCI machine learning data set [5], comprising 458 benign items (class 1) and 241 malignant items (class 2), each item being a vector of 9 real-valued numbers. The values were normalised to lie within the [0,1] interval.

5.1 Tuning the System

To tune the organism for this data set several settings were employed. Each experiment was repeated 20 times. Each run consisted of 3000 iterations with randomly picked data presented each iteration. Class 1 is treated as the "normal" class of data and class 2 is treated as "abnormal", from which one data item is introduced on average every 25 iterations (these values are taken from [3] to enable comparison), i.e. with a probability of 1/26.

All experiment settings involve a universe, an organism, a time system, an equal amount of waste recycling, absorption, cell recycling, expelling system varying in the experiments (see data introduction rate in Table 1), 250 data systems in experiments 1 to 12 and respectively 500, 750, 1000, 1250 and 1750 in experiments 13, 14, 15, 16 and 17. Each data system contains a decay and a split system. The organism initially contains 5 adhesion surfaces.

Table 1 shows the results of various tunings that were used. These results are computed by discarding the early computations (we discarded here the first 50000 computations) during which the organism grows to an adult (stable) state and stopping the experiments when the flow of data ends (thus not permitting organism's death from starvation).

From the first four experiments we can observe that the data similarity threshold has a significant impact on the performance of the program. The bigger the threshold, the less benign items are left unbound but the more malign cells can potentially bind somewhere. A threshold of 0.4 was then used for the next experiments.

Experiments 5, 6, 3 and 7 show that increasing the lifespan lowers the misclassification of class 1 data whilst slightly increasing the one of class 2. This parameter should thus be tuned depending on the priorities in the potential application (i.e. class to be most precisely detected).

Table 1. This table shows in percentage the average and standard deviation of data from each class creating a danger signal (false positive for class 1 and true positive for class 2) for various setups. Parameters are respectively in order: new data introduction rate (per cycle), data comparison threshold τ, cell's lifespan, and amount of growth systems. Experiments 1–4 investigate the effect of varying τ, experiments 5,6,3,7 investigate the effects of varying lifespan, experiments 8–11,3,12 investigate changing the amount of growth systems, experiments 3,13–17 investigate varying the data introduction rate.

Exp	Rate	τ	Lifespan	#Growth	Class 1		Class 2	
					mean	stddev	mean	stddev
1	1	0.2	15	100	22.70	1.27	99.87	0.39
2	1	0.3	15	100	11.54	0.76	99.63	0.69
3	1	0.4	15	100	7.56	0.51	98.72	1.01
4	1	0.5	15	100	5.39	0.67	96.19	1.69
5	1	0.4	5	100	9.16	1.11	99.20	1.17
6	1	0.4	10	100	7.85	0.48	99.06	1.14
7	1	0.4	20	100	7.23	0.41	98.73	1.06
8	1	0.4	15	5	14.40	0.81	99.91	0.42
9	1	0.4	15	10	8.98	0.57	99.61	0.62
10	1	0.4	15	25	7.85	0.59	99.53	0.87
11	1	0.4	15	50	7.62	0.40	98.95	0.90
12	1	0.4	15	150	7.62	0.50	98.44	1.00
13	2	0.4	15	100	6.99	0.59	98.56	1.36
14	4	0.4	15	100	7.01	0.68	97.88	1.34
15	8	0.4	15	100	7.48	0.94	98.88	1.24
16	16	0.4	15	100	6.85	0.81	98.33	2.39
17	24	0.4	15	100	7.24	0.90	98.29	2.21

Experiments 8 to 11, 3 and 12 show that, similarly to lifespan, increasing the amount of growth systems better classifies class 1 and lesser classifies class 2.

Experiments 3 and 13 to 17 show that varying the data introduction rate does not have a significant impact on the classification accuracy.

Comparing these results with the ones from [3] and looking at the best setups, we clearly outperform their results. It is interesting to notice that the best setups here use a threshold of 0.4 against 0.2 in [3]. It seems that for this study, having a low threshold in a deterministic program as in [3] is better whilst in a stochastic approach such as SC a larger threshold works better.

5.2 Looking into the Organism

This section investigates what can be learnt from the organism's inner state over time. It is expected that observing the organism from within should reveal information about the data set. To be an effective and useful tissue algorithm, suitable for use with AIS algorithms, the organism should organise itself in a stable pattern reflecting the data distribution. If for instance the data stream contains three distinct sets we expect to observe three distinct clusters of cells.

The relevant values to observe along the computation are the amount of cells, clusters and danger signals. Similarly to a real case on-line program execution, Figure 5 shows the state over time of an organism during a run of 5000 samples using the configuration of experiment 3. Again, the computations corresponding to the organism's early life (here the first 100000 iterations) are discarded in order to focus on the mature aspect of the organism.

Results from Table 1 already provide insights regarding the inner shape of the organism (i.e. its inner organisation). Class 2 (malign) items are very well identified which means that class 2 data are not easily aggregated to other data (otherwise they could not be well detected). Therefore class 1 (benign) data only is actually clustering and it is thus expected to observe on average one main cluster all along the program execution.

Figure 5 shows that the organism has a constant amount of cells in spite of constant cellular death. The organism therefore shows homeostatic behaviour. Danger signals are regularly emitted and represent the detected supposedly malign cells that could then be used by an AIS algorithm. The curve at the bottom

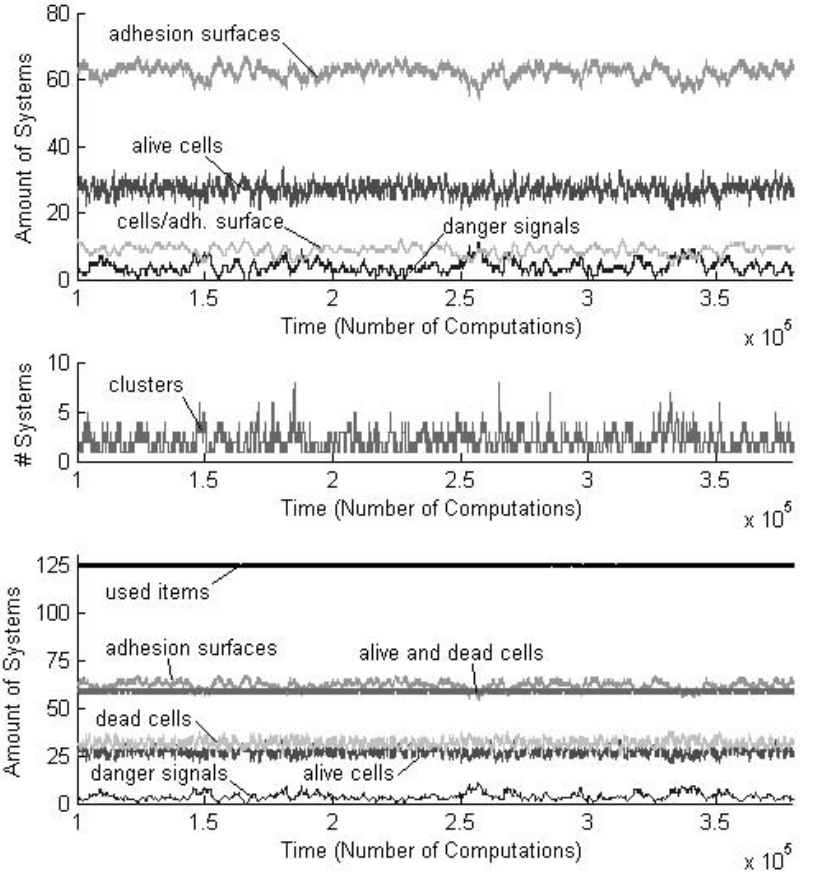


Fig. 5. Organism's inner organisation over a run of 5000 samples

shows the amount of clusters over time (in a smaller scale along the Y-axis for clarity). We can observe that as expected the organism keeps settling down into one cluster. New clusters are constantly created with the appearance of new adhesion surfaces but quickly these new clusters bind to the main one.

As the organism is designed to grow to match the data rate, such a program is therefore able to cope with various (unexpected) parameter changes, self-(re)organising with the data flow, and providing information over time regarding detected potentially abnormal data items. When used in conjunction with an artificial immune algorithm, the good accuracy of detection (albeit with a high false positive rate), and the automatic organisation of similar data into clusters should enable excellent performance overall. While temporal information is currently lost because of the stochastic computation of SC, this could be added as additional features of data items, enabling the clustering according to similar timings in addition to data values.

One advantage of SC is the simplicity of modelling new stochastic systems, so an immune algorithm could be added to this model by simply adding two or three new types of system (e.g. B-cell, T-cell, antibody) that would then automatically interact with the existing tissue systems. Another valuable advantage of using SC in our approach is the fault-tolerance and self-repair ability an SC model can naturally have, as investigated in [16]. Having robust software can indeed be an important feature in network security to ensure the program can survive even severe damage provoked for instance by hacking.

6 Conclusion

In this paper we introduced the notion of artificial metabolism using systemic computation to create an organism for clustering data that is suitable for an artificial immune system. This work is inspired by Matzinger's Danger Theory and uses the notion of danger signals. Starting from scratch and working on-line our organism is able to cluster data according to its similarities and can provide danger signals when cells die in an abnormal way for the current organism. Our organism proved to be able to detect anomalous UCI Breast Cancer data with better accuracy than in previous work [3]. The study of the evolution over time of our organism showed that its inner organisation reflects the data distribution of the current flow. Also, previous results have shown that SC programming is very robust, with programs easily showing fault-tolerance and self-repair abilities even when undergoing severe damage [16].

References

1. Aickelin, U., Greensmith, J.: Sensing Danger: Innate Immunology for Intrusion Detection. Elsevier Information Security Technical Report, pp. 218–227 (2007)
2. Matzinger, P.: Tolerance, Danger and the Extended Family. Annual Reviews in Immunology 12, 991–1045 (1994)

3. Bentley, P.J., Greensmith, J., Ujjin, S.: Two Ways to Grow Tissue for Artificial Immune Systems. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 139–152. Springer, Heidelberg (2005)
4. Bentley, P.J.: Systemic computation: A Model of Interacting Systems with Natural Characteristics. Int.J. Parallel, Emergent and Distributed Systems 22(2), 103–121 (2007)
5. Breast Cancer Wisconsin (Diagnostic) Data Set, Creator: Wolberg, W. H., Donor: Mangasarian, O., UCI Machine Learning Repository (1992), http://archive.ics.uci.edu/ml/
6. Tempesti, G., Roggen, D., Sanchez, E., Thoma, Y.: A POEtic Architecture for Bio-Inspired Hardware. In: Proc. of the 8th Intl. Conf. on the Simulation and Synthesis of Living Systems (Artificial Life VIII), pp. 111–115. MIT Press, Cambridge (2002)
7. Thoma, Y., Tempesti, G., Sanchez, E., Moreno Arostegui, J.-M.: POEtic: an electronic tissue for bio-inspired cellular applications. BioSystems 76, 191–200 (2004)
8. Wallenta, C., Kim, J., Bentley, P.J., Hailes, S.: Detecting Interest Cache Poisoning in Sensor Networks using an Artificial Immune Algorithm. Journal of Applied Intelligence (to appear, 2008)
9. von Neumann, J.: The theory of self-reproducing automata. Univ. of Illinois Press, Champaign (1966)
10. Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Champaign (2002)
11. Holland, J.H.: Emergence, From Chaos to Order. Oxford University Press, Oxford (1998)
12. Adamatzky, A.: Computing in Nonlinear Media and Automata Collectives. Institute of Physics Publishing, Bristol (2001)
13. Arvind, D.K., Wong, K.J.: Speckled Computing: Disruptive Technology for Networked Information Appliances. In: Proc. of the IEEE Intl. Symposium on Consumer Electronics (ISCE 2004), pp. 219–223 (2004)
14. Le Martelot, E., Bentley, P.J., Lotto, R.B.: A Systemic Computation Platform for the Modelling and Analysis of Processes with Natural Characteristics. In: Proc of 9th Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 2809–2816. ACM Press, New York (2007)
15. Le Martelot, E., Bentley, P.J., Lotto, R.B.: Exploiting Natural Asynchrony and Local Knowledge within Systemic Computation to Enable Generic Neural Structures. In: Proc of 2nd International Workshop on Natural Computing (IWNC 2007) (2007)
16. Le Martelot, E., Bentley, P.J., Lotto, R.B.: Crash-Proof Systemic Computing: A Demonstration of Native Fault-Tolerance and Self-Maintenance. In: Proc of 4th IASTED International Conference on Advances in Computer Science and Technology (ACST 2008). ACTA press (2008)

An Immune System Based Multi-robot Mobile Agent Network

W. Wilfred Godfrey and Shivashankar B. Nair

Indian Institute of Technology Guwahati, Guwahati, India
{w.godfrey,sbnair}@iitg.ernet.in

Abstract. A biological immune system comprises components that circulate within the body and defend it from attacks by pathogens. Its artificial counterpart has found applications in a wide spectrum of domains including Robotics. The cells in an immune system can be viewed to be analogous to mobile agents that migrate from one node (organ) to another. Mobility of such agents provides the framework for movement of components within a system. In this paper we describe the modeling of a mobile agent based robotic system that works on the principles of an artificial immune system. The mobile agents form the B-cell equivalents, generating rule-sets (antibodies) that help the robot to overcome a problem (antigen).

Keywords: Mobile Agent, Artificial Immune System, Robotics.

1 Introduction

Natural processes are characterized by their complex dynamics and interactions [1]. The complexities involved in these processes produce behaviors that are non-trivial and highly sophisticated. The Biological Immune System (BIS) is a typical example of such a process. It provides high level biological processing capabilities and acts independently [2]. The properties of the biological immune system are highly appealing and have diverse applications in the world of information processing [2, 3, 4, 5].They also have found their way into the domain of robotics. Besides, the recent surge in the use of robots has forced many a researcher to employ IS based algorithms for robot control. Ishiguro et al have applied IS principles in robotics mainly for behavior arbitration [6, 7, 8, 9] and for gait control of walking robots [10, 11, 12]. Dong and Kwee [13] describe an immune network theory based co-operative control of autonomous mobile robots, termed Distributed Autonomous Robotic System in which the desired effect is produced as an emergent behavior of the robots. Hart [14] has used the IS based approach to create '*growing up*' of rules for accomplishing complex tasks. More IS based approaches used in robotics are demonstrated in [15, 16]. The BIS has features that are distributed, robust and easily adaptable. They are thus well suited for controlling multiple robots. An IS based multiple autonomous mobile robot control is discussed in [13]. Mobile agents are autonomous and goal-driven. They readily provide distributed computing functionalities and have added advantages as mentioned in [17].This makes them a novel and effective design philosophy. Mobile agents have found their way into the robotic domain too [16, 18, 19, 20]. Shin and Lee [16] propose a multi-robot mobile agent method for securing the system

P.J. Bentley, D. Lee, and S. Jung (Eds.): ICARIS 2008, LNCS 5132, pp. 424–433, 2008.

reliability and safety. Cragg and Hu [19, 20] elaborate on how the mobile agents augment a distributed computing system. These robots can share knowledge and information to achieve complex tasks.

While most of these systems deal with concepts for tasks based on BIS, we are not aware of any attempt to use mobile agents to imitate B- or T- cells to emulate a real immune system for a family of robots. In this paper we describe a multi-robot system derived on the principles from BIS and based on the philosophy of mobile agents.

Motivations that prompted the use of mobile agents -

1. Agent mobility increases the potential for sharing information with other agents and hence with other robots being serviced by them.
2. Robots can pass on information, learned through experiences in their respective environments, to other robots via mobile agents.
3. Fine tuning of the information from different robots could be performed by these agents. Further their interaction with other agents could be viewed as an immune network [2] that shares and refines information.
4. Robot-specific mobile agents could be summoned to a robot mimicking aspects of chemical signaling in the immune system world.

While section 2 describes the motivation behind this system, section 3 describes an overview of the system along with a list of metaphors used. Section 4 reveals the implementation aspects and the analysis of results and finally section 5 concludes the paper.

2 Overview of Biological Immune Systems (BIS)

Reproduction in living organisms occur by copying the molecular instruction manual, called genes and passing it on to the next generation. Family traits are thus passed from one generation to the succeeding ones. The collection of all the genes in a living organism is called its genome [21]. Human beings have fewer than 10^5 genes in their entire genome. An antibody is an element of the immune system. The genetic material to produce an antibody molecule is stored in the component libraries. Random selection of the genetic material from these libraries results in the production of an antibody molecule [22].

The immune system serves to protect the body against foreign organisms. The innate immune system is the first line of defense against a foreign attack. After an interim period, it initiates the adaptive immune system. This largely comprises of the B- and T- type cells produced mostly in the bone marrow and the thymus. These cells are generated as precursor cells in the bone marrow and migrate to the thymus where they eventually mature before being released into the blood stream. The T-cells are known to stimulate their B- counterparts when an antigen is detected. An antigen could be any substance that induces an immune response. This causes the B-cell to release large numbers of antibodies with a single specificity that tackles the antigen.

The behavior of the adaptive immune system to produce a large number of single specific antibodies over a huge range of antigens is explained by clonal selection theory [2]. Binding of an antigen with antibody receptors results in the promotion of new antibodies. This is the core of adaptive immune response. The process of refinement of the antibody specificity affecting the genes occurring in the B-cell is called somatic

mutation. A unique shape on the surface of an antigen which triggers an antibody response is known as epitope. The portion of antibody molecule that recognizes an epitope is called paratope.

In the domain of robotics, immune systems tackle the two basic problems of constructing an arbitration mechanism and preparing appropriate competence modules which occur in behavior based robot systems [23].

3 BIS Based Mobile Agent Multi-robot System

We describe a multi-robot networked system based on [24, 25, 26]. The system comprises nodes, mobile agents and robots as depicted in Figure 1.

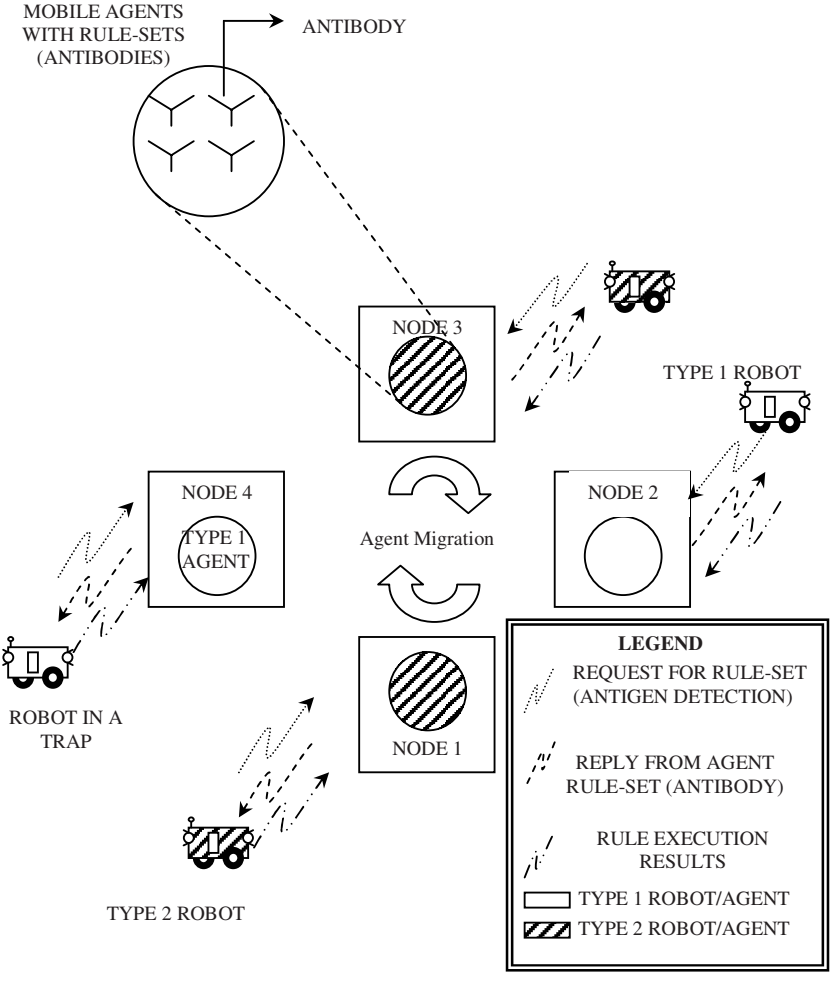


Fig. 1. Multi-robot system Architecture

A *node* acts as a placeholder for the agents. Robots communicate with the agent through the node they are tethered to. Agents are mobile and thus are capable of movement from one node. We have assumed that they move in a round robin manner. Robots depend on the agents to provide information when they encounter situations that they cannot tackle on their own.

The *type* of a robot is defined by its configuration, the task assigned and its operating environment. The *type* of an agent is defined by the family of robots it can provide services to. An agent of one type cannot respond or support a robot of a different type.

The robots are autonomous but not totally intelligent in themselves. For instance a robot may encounter a situation it cannot comprehend such as an obstacle in front. The complexity of a situation that a robot faces is measured in terms of *pain*. Pain symbolizes the discomfort the robot faces in a situation. If the pain level increases beyond a certain threshold, the robot detects an antigenic invasion and signals for aid from the agent. Under such conditions the agent at the respective node is contacted which in turn responds by providing the necessary information, provided it is of the compatible type. The information that the robot receives from the agent is in the form of a set of rules termed the *rule-set*. By executing the rules within, it is hoped that the robot will be able to overcome the antigenic invasion and reduce the pain to a value below the threshold. The rule-set thus acts as an antibody (provided by a B-cell viz. the mobile agent) trying to defend the robot The robot utilizes the rule-set to possibly overcome a problematic situation and relays its effects back to the agents. The agent in its turn analyzes this information and modifies the rule-sets if required. The robot is programmed to take up pain values beneath a threshold and fend for problems with lower intensities. This is analogous to the innate immune system.

3.1 Pain Function

One significant feature of the system is self-preservation. The term self-preservation denotes the instinct which helps an animal to survive fear and pain. In the current scenario the self preservation of the system is defined as the procedure for detecting discomfort (pain) and increasing the level of comfort. Discomfort or pain is expressed as a function of the various sensory perceptions of a robot. The sensory perceptions may be the internal state of the robot or the environmental conditions perceived by the robot. The pain function can be modeled based on the behavior desired from a robot. Thus,

Pain = Pain due to change in internal state + Pain due to external Percepts
= Battery charge + Distance from the obstacle as sensed by the Sensor

Pain = w_B*P (B) +w_L *P (L)

P (B) =Battery sensor value = 100 if battery sensor value is 0
90 if battery sensor value is 50
75 if battery sensor value is 75
0 if battery sensor value is 100

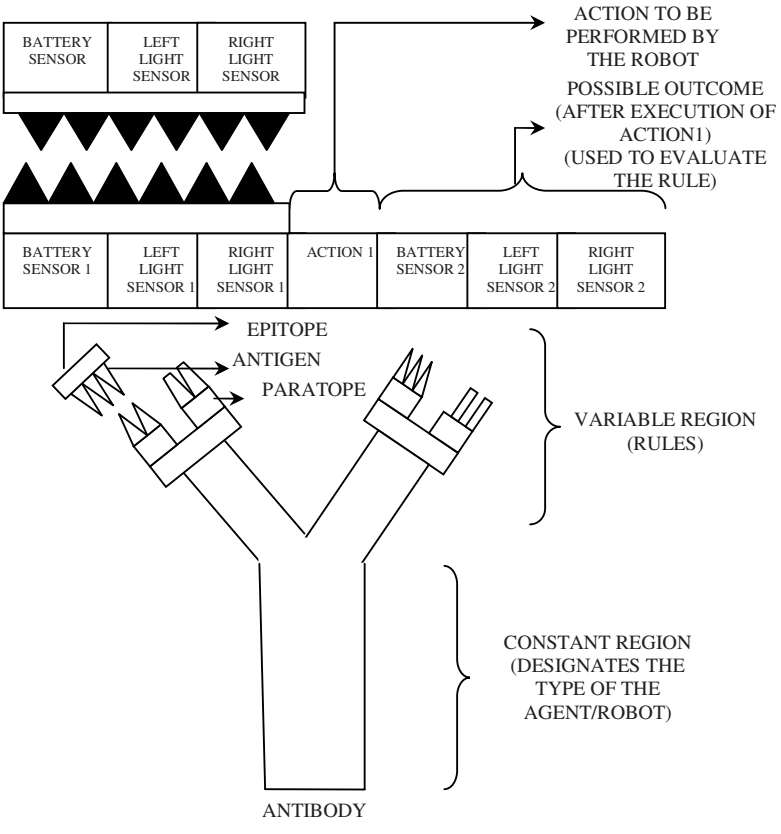


Fig. 2. Structure of a rule-set and example of Antibody-Antigen association with the related metaphors

In our experiments, these values were calculated by fitting the values on the left on to the values on the right. This was carried out by interpolating the corresponding polynomial using Lagrange's method. The distance sensor used shows a maximum value when it is close to the obstacle and a value of 0, when far away. Thus,

$$P(L) = \text{Pain perceived via the distance sensor} = \text{Distance sensor value}$$

w_B and w_L are weights which are chosen such that the effects of both the percepts are held in balance while calculating the pain.

3.2 Rules and Rule-Sets

Rules form the basic building block of the antibody (rule-set). Rules are thus synonymous to genes. Since genes can be formed only over several generations, we made a robot situated in an environment to randomly discover a large number of actions to circumvent obstacles. The actions taken by this gene generating robot were rated based on their initial (before taking the action) and final (after execution of the action)

pain values and ranked based on how well each action could reduce the pain levels. The best ranked rules formed the initial gene library.

$$\text{The change of pain } (\Delta P) = \text{Pain before action} - \text{Pain after action}$$

$$= P_b - P_a.$$

Gene library contains rules, whose ΔP is positive, and above threshold. The threshold (σ) is found out by normalizing the ΔP values and by calculating the mean over those values. This same value is also used for triggering the robot to seek help from the agent or its rule-set. A rule-set is framed by randomly picking up n rules from the rule-library each of which produces a significant reduction in pain value and each of which have a well dispersed sensor range. This hopefully ensures a uniform distribution while randomly selecting rules that form a rule set. Figure 2 depicts a typical rule comprising seven fields viz. the battery sensor value indicative of the charge left on the battery, the left and right sensor values, the *action* to be taken when these sensory conditions are reported and three fields that designate the values of the sensors that could be expected after execution of the action (as reported by the gene library). Thus the latter three sensor fields contain values which the initial gene-generating robot obtained after performing the relevant action defined in the action field. These fields are used to evaluate how well the current robot's environment and actions match those of the robot used to generate the initial genes. In our scenario we have taken four rules to form a rule-set (antibody). As can be seen from Figure 2, each variable limb of the Y shaped antibody has two paratopes. Each paratope represents a rule. The constant region specifies the type or family of robots, this antibody can be used for. Contrary to the biological equivalent, this antibody provides specificity to four different antigens or sensory conditions conforming to the four rules. The system as a whole thus endeavors to refine the rules so that the *self* is preserved.

Whenever the robot executes a rule it preserves the next state of the three sensors. This information is sent back to the agent which evaluates the performance of the rule (antibody) using the equation –

The effectiveness of a triggered rule is given by

$$E_R = \frac{\text{Number of times the rule was successful in reducing the pain.}}{\text{Number of times the rule was triggered}}.$$

3.3 Metaphors Used in the System

The multi-robot system is modeled based on the Biological Immune system world. The mechanism to produce the initial rules using a gene generating robot is prototyped based on the Thymus and Bone-marrow model. The manner of segregation of useful rules and the creation of a rule library is similar to that of cell and non-cell differentiation performed in the thymus and bone marrow. The initial set of rules resembles the genome and the rules match the genes.

The robots are modeled as T-cells, who stimulate the immune response on detection of the presence of antigens in their respective environment. Agents are the B-cells induced into action by the T-cells. This whole situation is akin to the antigen stimulating a macrophage and the mechanism involving the secretion of the antibodies. The

rule sets constituting the antibodies, are then used by the robot and graded based on their effectiveness in the real world. Effectiveness is a measure of the amount of pain-reduction. Their effectiveness is relayed back to the agents as feedback and the agents analyze and process the feedback. This process of purging less effective rules is akin to the negative selection. With more efficient rules circulating within the network, more robots benefit from them leading to a faster secondary response. Table 1 lists the AIS metaphors used in this multi-robot mobile agent system.

Table 1. Metaphors used in the multi-robot system

Robot System	Genetics and Immune System
Multi-robot system+Agent system	Adaptive immune system
Robot	B-Cells
Mobile Agents	T-Cells
Initial Set of rules	Genome
Pruned Rules	Gene library
Rules	Genes
Pain function	Gene Fitness
Rule Generation	Thymus and Bone Marrow
Rule purging	Negative Selection
Robot Action rule set	Antibody
Pain	Self-preservation
Pain value above threshold	Antigens(Stimulus)
Mobile Agent system	Flow of Cells
Battery	Lifetime of the cell

4 The Prototype

We used Java J2SE 1.4.2 and JADE 3.4.1(mobile agent platform to host the mobile agents) coupled with Lego Mindstorm® robots for testing the prototype of the system. The robots were equipped with an in-built battery sensor and two distance sensors. The mobile agent library in JADE provides functions for creation, destruction, cloning, migration and storage of agents. The agent programs are written in Java and the robot used LeJOS®, a tiny Java based OS for Lego Mindstorms® RCX.

The environment contained rectangular areas with three obstacles randomly placed within and an area that simulates charging of the battery in one corner of the environment.

The robot was capable of four primitive actions viz. *move forward*, *move left*, *move right* and *go back*. The robot communicated through an infrared tower with the mobile agents running on PCs that acted as nodes. Agents were created and destroyed manually. They live and continually migrate. The *type* of the agent was defined during the agent creation process. As the agents migrate from one node to another, they served the robots which were of a similar type as and when required.

4.1 Experiments and Results

We observed that mobility plays a vital role in keeping the cells alive and warding off the Antigen, When the robot encounters an obstacle its pain value increases, (Antigen

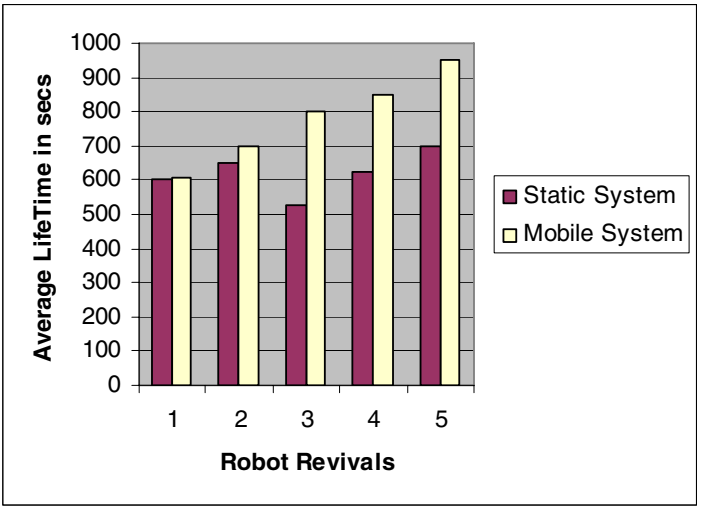


Fig. 3. Comparison of robot life times

detection) resulting in Queries being sent to the Agent. If the agent is able to provide rules that can efficiently avoid the obstacle, it leads to conserving the battery charge, effectively increasing the lifetime of the robot.

The experimental scenario comprised 2 nodes each of which had one agent with no mobility and 2 robots. Both robots were equipped with identical sensors as mentioned earlier and were assigned the same task of obstacle avoidance. The environments in which the robots were situated were also similar in nature. The maximum amount of time for which the robots lived (battery life>0) was found. The robots were again charged and the experiment was repeated constituting a robot revival. The same experiment was carried out with the agents being conferred mobility. The graph in Fig. 3 shows a marked change in the average lifetimes of the two robots when the agents are mobile. This can be attributed to the mobile agents transferring experiences gained amongst each other.

5 Conclusions

This paper proposes a model wherein mobile agents mimic lymphocytes. The B-cells act only when they receive a stimulus from the T (helper) cells. In this model too the mobile agents act only on receiving a stimulus from the robot. Flow of these mobile agents in network populated by robots facilitates sharing knowledge thereby increasing the effective lifetime of the entire network of robots. Experimental results clearly demonstrate performance gains in such a system when mobile agents are used. It is envisaged that a complete implementation of the model described herein will result in the network exhibiting an emergent behavior.

References

[1] Jonker, C., Treur, J.: Agent Oriented modeling of the dynamics of biological organisms. Journal of Applied intelligence 27(1), 1–20 (2007)
[2] de Castro, L.N., Timmis, J.: Artificial Immune Systems: A new Computational Intelligence Approach. Springer, London (2002)
[3] Carvalho, D.R., Freitas.: An Immunological Algorithm for discovering small-disjunct rules in data-mining. In: Proceedings of the Genetic and Evolutionary Computation Conference (2001)
[4] Forrest, S., Hofmeyr, S.A., Somayaji.: Computer Immunology. Communications of the ACM 40(10), 88–96 (1997)
[5] de Castro, L.N., Zuben, F.J.: aiNet: An Artificial Immune Network for Data Analysis. In: Data Mining: A heuristic Approach, ch. 12, pp. 231–259. Idea group Publishing, USA
[6] Patricia, V., de Castro, L.N., et al.: An Immune Learning Classifier Network for Autonomous Navigation. In: Second international conference on Artificial Immune System (2003)
[7] Ishiguro, A., Watanabe, Y., Kondo, T., Shirai, Y., Uchikawa, Y.: A Robot with a Decentralized Consensus-making Mechanism Based on the Immune System. In: Proceedings of ISADS 1997, pp. 231–237 (1997)
[8] Ishiguro, A., Shirai, Y., et al.: Immunoid: An architecture for behavior arbitration based on the immune networks (1996)
[9] Ishiguro, A., Watanabe, Y., Kondo, T., Uchikawa, Y.: Immunoid: A Robot with a Decentralized Behavior Arbitration Mechanisms Based on the Immune System. In: Proceedings of ICARCV 1996, vol. 2, pp. 1600–1605 (1996)
[10] Ishiguro, et al.: Moderationism in the immune system: Gait acquisition of a legged robot using the metadynamics function. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, pp. 3827–3832 (1998)
[11] Ishiguro, A., Ichikawa, S., Uchikawa, Y.: A gait acquisition of a 6-legged robot using immune network. In: Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, vol. 2, pp. 1034–1041 (1994)
[12] Ishiguro, et al.: Gait coordination of hexapod walking robots using mutual coupled immune networks. In: IEEE International Conference on Evolutionary Computation, vol. 2, pp. 672–677 (1995)
[13] Lee Dong, W., Sim, K.: Artificial Immune network-based cooperative control in collective autonomous mobile robots. In: 6th IEEE International Workshop on Robot and Human Communication (1997)
[14] Hart, E., Ross, P., Webb, A., et al.: A role for immunology in "Next Generation" robot controllers. In: Second international conference on Artificial Immune System (2003)
[15] Henry, L., Wong, V.: Immunologic control framework for automated material handling. In: Second international conference on Artificial Immune System (2003)
[16] Sung-Oog, S., Jug-Ooo, L., Kwon, B.D.: A mobile agent based multi-robot design method for high-assurance. In: 10th IEEE High Assurance Systems Engineering Symposium, pp. 389-390 (2007)
[17] David, C., Harrison, C., Kirshenbaum, A.: Mobile agents: Are they a good idea? In: Second International Workshop on Mobile Object Systems - Towards the Programmable Internet (1996)
[18] Cragg, L., Hu, H.: Application of Mobile agents to robust teleoperation of internet robots in nuclear decommissioning. In: IEEE conference on Industrial technology (2003)
[19] Cragg, L., Hu, H.: A multi-agent system for distributed control of networked mobile robots (December 2005)

[20] Cragg, L., Hu, H.: Mobile Agent approach to networked robots. The International Journal of Advanced Manufacturing Technology 30(9-10), 979–987 (2006)
[21] The new genetics. U.S Dept of Health and Human Services. National Institute of General Medical Sciences
[22] Hightower, R., Forrest, S., Perelson, A.S.: The evolution of emergent organization in immune system gene libraries. In: Proceedings of Sixth International Conference on Genetic Algorithms, pp. 344-350 (1995)
[23] Liu, J., Wu, J.: Multi-agent robotic systems, 1st edn. CRC press, Boca Raton (2001)
[24] Nair, S.B., Toppo, N.: A Framework for Sharing Intelligence among Mobile Robots on a Network. In: Proceedings of the 2nd IASTED International Multi-Conference on Automation, Control and Information Technology, pp. 93-98 (2005)
[25] Jerne, N.K.: The generative grammar of the Immune system. Bioscience Reports 5, 439–451 (1985)
[26] Nair, S.B., Pradeep, K.V.D., Saravanan, M.: A Communication Protocol for a Mobile Adhoc Network of Robots. In: The International Conference on Emerging Applications of IT (2006)

Author Index